Oracle Fusion Cloud HCM

HCM Data Loading Business Objects

24A

Oracle Fusion Cloud HCM
HCM Data Loading Business Objects

24A

F88707-05

Author: Oracle Human Resources Information Development Team

Contents

Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

Get Help in the Applications

Use help icons to access help in the application. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons.

Get Support

You can get support at *My Oracle Support*. For accessible support, visit *Oracle Accessibility Learning and Support*.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at *Oracle University*.

Join Our Community

Use *Cloud Customer Connect* to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest *ideas* for product enhancements, and watch events.

Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program*. Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to *oracle_fusion_applications_help_ww_grp@oracle.com*.

Thanks for helping us improve our user assistance!

1 Introduction to Loading Business Objects using HCM Data Loader

Loading Business Objects: Overview

This documentation describes the guidelines and examples for loading specific business objects in the Oracle HCM Cloud.

Introduction

HCM Data Loader is a powerful tool for bulk-loading and maintaining data. It supports file-based upload of business objects, components, and component attributes using a flexible, delimited file. It's used for data migration, ongoing maintenance of HCM data, and coexistence scenarios, where core HR data is uploaded regularly.

Oracle HCM Cloud business objects can be complex and usually hierarchical, allowing for multiple child records to exist for a business object.

Each component within a business object hierarchy supplies multiple attributes. These attributes are references to other objects, some of which are lookup-validated, and most are optional. Some are required only for new records, while others are always required.

See *Sources of Business-Object Information* to understand the structure, supported key systems, data formats, required attributes, and the business rules applicable for the business object when loading a business object or a component.

And here's a tutorial on *View Business Objects* that helps you review business objects, generate template files, and download business object overview spreadsheets.

Note:

HCM Data loader doesn't support loading of third party organizations. If you want to load any third party organization, take a look at the **About Import Management** section and the **Import Your Organization Data** help topic in the "**Using Customer Data Management (CDM) for CX Sales and B2B Service**" guide on Oracle Help Center.

Related Topics

- How do I import organization data?
- How do I import data?

2 Loading Setup Objects

Overview of Loading Setup Objects

This chapter covers the guidelines and examples for loading these setup objects in the Oracle HCM cloud: absence cases, actions, action reasons, calendar events, checklist templates, document types, document delivery preferences, extended lookup codes, name formats, person types, resource exceptions, and time and labor.

The guidelines describe how to load these objects using HDL.

Guidelines for Loading Absence Cases

You use absence cases to group related absences, typically those having the same cause. For example, a worker may take two periods of sickness absence after sustaining an injury at work.

In this case, you can associate the two absences with the same absence case for ease of management. This topic describes how to load Absence Case objects for a worker using HCM Data Loader. Use the AbsenceCase.dat file

Absence Categories

If you associate an absence case with an absence category, then all absences in the case must have the same category. If you specify no category for the absence case, then you can associate any type of absence with the absence case. You define absence categories using the **Manage Absence Categories** task in the Absence Administration work area.

Loading Absence Cases

These records must exist in Oracle HCM Cloud before you can create an absence case that refers to them:

- Absence category, if you're using categories
- Worker and work relationship
- Worker assignment, if absences are specific to the assignment
- Absence records

Deleting Absence Cases

You can delete Absence Case objects using HCM Data Loader. To identify the record to be deleted, supply the **AbsenceCaseCode** attribute.

Examples of Loading Actions

Actions classify changes to data such as employment and compensation records. When you create or update a record, the action value specifies the reason for the change. For example, the associated action for an assignment change may be promotion or transfer.

Action reasons, which are optional, provide additional information about a change. The Action Reason Usage component of an Actions object specifies the relationship between an action and an existing action reason. This topic shows how to load Actions objects and their Action Reason Usage components using HCM Data Loader.

Creating Actions

This example Actions.dat file creates an Actions component and associates it with an Action Reason Usage component. Both components are identified by source keys.

```
METADATA|Actions|SourceSystemOwner|SourceSystemId|ActionCode|ActionName|ActionTypeCode|StartDate|EndDate
MERGE|Actions|VISION|VISION_ACT_PROM|VISION_PROM|Vision Promotion|EMPL_PROMOTION|2000/01/01|
METADATA|ActionReasonUsage|SourceSystemOwner|SourceSystemId|ActionCodeId(SourceSystemId)|
ActionReasonId(SourceSystemId)|StartDate|EndDate
MERGE|ActionReasonUsage|VISION|VISION_PROM_PERF|VISION_ACT_PROM|VISION_PERF|2000/01/01|
```

Loading Translated Action Names and Descriptions

Supply action names and descriptions in the language of the user who's loading them. You supply an ActionsTranslation.dat file to translate action names and descriptions into other enabled languages once the actions exist. This example ActionsTranslation.dat file translates an action name and its description. The action is identified by its source key.

```
METADATA|ActionsTranslation|SourceSystemOwner|SourceSystemId|Language|ActionName|Description
MERGE|ActionsTranslation|VISION|ACT_VISION_PROM|FR|Performance|Changement est survenu en raison de la
 performance des employes
```

Related Topics

- Examples of Loading Action Reasons

Examples of Loading Action Reasons

Action reasons, which are optional, provide additional information about an action. For example, the reason for promoting a worker may be good performance or seniority. This topic shows how to load action reasons using HCM Data Loader.

Creating Action Reasons

This example ActionReasons.dat file creates two Action Reasons components. The components are identified by source keys.

```
METADATA|ActionReasons|SourceSystemOwner|SourceSystemId|ActionReasonCode|ActionReason|StartDate|EndDate
MERGE|ActionReasons|VISION|AR_VISION_PERF|VISION_PERF|Performance|2000/01/01|4712/12/31
```

```
MERGE|ActionReasons|VISION|AR_VISION_TXFR|VISION_TXFR|Internal Transfer|2000/01/01|
```

Loading Translated Action-Reason Names

Supply action-reason names in the language of the user who's loading them. You supply an ActionReasonsTranslation.dat file to translate action-reason names into other enabled languages once the action reasons exist. This example translates the names of existing action reasons. The Action Reasons components are identified by their source keys.

```
METADATA|ActionReasonsTranslation|SourceSystemOwner|SourceSystemId|ActionReason|Language
MERGE|ActionReasonsTranslation|VISION|AR_VISION_PERF|Accomplissement|FR
MERGE|ActionReasonsTranslation|VISION|AR_VISION_TXFR|Transfert interne|FR
```

Deleting Action Reasons

You can delete Action Reasons components using HCM Data Loader. This example ActionReasons.dat file deletes two action reasons. It identifies the components using source keys.

```
METADATA|ActionReasons|SourceSystemOwner|SourceSystemId
DELETE|ActionReasons|VISION|AR_VISION_PERF
DELETE|ActionReasons|VISION|AR_VISION_TXFR
```

Related Topics

- Examples of Loading Actions

Guidelines for Loading Calendar Events

A calendar event is an event, such as a public holiday or plant closure, on which working time may be affected. Calendar events are optional. You specify the geographical or organizational hierarchy to which workers affected by the calendar event must belong.

The hierarchy must exist before you load the associated calendar events. This topic describes aspects of calendar events that you must understand to load them successfully using HCM Data Loader. It also provides examples showing how to load calendar events.

Calendar Event Lookup Categories

Calendar events have categories, which are defined in the PER_CAL_EVENT_CATEGORY lookup type. This lookup type has one delivered value, PH, for public holidays. If you use additional categories, then you must define them before you load calendar events. In the Setup and Maintenance work area, use the following:

- Functional Area: Workforce Information
- Task: Manage Availability Lookups

Calendar Event Coverage

The Calendar Event Coverage component identifies the branch of the geographical or organizational hierarchy to which the associated calendar event applies. You identify the top node of the hierarchy branch. The event applies to that node and its child nodes. You can also:

- Exclude from the coverage individual nodes that appear in the specified hierarchy branch.
- Override the calendar event name and its category for individual nodes in the hierarchy.

In a geographical hierarchy, the calendar event applies to all workers with assignments in the locations that you include in the calendar event coverage. In an organizational hierarchy, the calendar event applies to all workers with assignments in the departments that you include in the calendar event coverage. The affected workers may also have work schedules assigned to them. In this case, the event applies to the workers only if you add it as a resource exception to the work schedule or work schedule assignment.

Creating Calendar Events

This example CalendarEvent.dat file creates both a public holiday for Christmas Day and a half-day event for elapsed work schedules. It identifies the calendar events by their source keys.

```
METADATA|CalendarEvent|Name|Description|Category|CoverageType|ShortCode|StartDateTime|EndDateTime|TreeCode|
TreeStructureCode|TreeVersionName|HalfDayForElapsed
MERGE|CalendarEvent|CPTAS6|||PH|G|TAS6|2017/07/11 08:00:00|2017/07/11 12:30:00|WFMTL_Global|
PER_GEO_TREE_STRUCTURE|WFMTL Bank Geography Version 1|Y
MERGE|CalendarEvent|Christmas Day 2017|||PH|G|XMAS2017|2017/12/25 08:00:00|2017/12/25 18:00:00|WFMTL_Global|
PER_GEO_TREE_STRUCTURE|WFMTL Bank Geography Version 1|N
METADATA|CalendarEventCoverage|ShortCode|CoverageScope|TerritoryCode|TreeStructureCode|TreeCode|
TreeVersionName
MERGE|CalendarEventCoverage|TAS6|I|US|PER_GEO_TREE_STRUCTURE|WFMTL_Global|WFMTL Bank Geography Version 1
MERGE|CalendarEventCoverage|XMAS2017|I|US|PER_GEO_TREE_STRUCTURE|WFMTL_Global|WFMTL Bank Geography Version 1
```

Deleting Calendar Events

You can delete a calendar event unless it's assigned to a work schedule. This example CalendarEvent.dat file deletes a calendar event. It identifies the calendar event by its source key.

```
METADATA|CalendarEvent|SourceSystemOwner|SourceSystemId
DELETE|CalendarEvent|VISION|XMAS2017
```

Related Topics

- Examples of Loading Resource Exceptions

Guidelines for Loading Checklist Templates

You use a checklist template to define a sequence of related tasks with multiple performers, such as those for onboarding a new worker.

For example, you can create a checklist template for new hires, with tasks for providing system access, issuing badges, allocating parking space, and so on. This topic describes how to create and maintain Checklist Template objects using HCM Data Loader.

Checklist Categories

The checklist category must exist in the CHECKLIST_CATEGORY lookup for the target environment. In the Setup and Maintenance work area, use the following to manage checklist categories:

- Functional Area: Workforce Information
- Task: Manage Checklist Lookups

The combination of checklist name and checklist category must be unique. **Enterprise Onboarding Step** checklists must either already exist in the target environment or be in the same .dat file as **Enterprise Onboarding** checklists that reference them.

Checklist Tasks

These rules apply to checklist tasks:

- Checklist task names must be unique for the checklist.
- If the current task has a preceding task, then the preceding task must be loaded before the current task can be created.
- When you create tasks of the type **Configurable Form**, the descriptive flexfield context must have been configured in the target environment.

Checklist Actions, Areas of Responsibility, and Eligibility Profiles

If you plan to use actions in the checklist template, then those actions must exist in the target environment.

If you plan to use areas of responsibility to identify task performers, then responsibility types must exist in the PER_RESPONSIBILITY_TYPES lookup in the target environment. Alternatively, you can use one of the values shown in this table.

Value	Meaning
ORA_LN_MGR	Line manager
ORA_WORKER	Worker
ORA_INITIATOR	Initiator

The default performer is the line manager.

If you plan to use eligibility profiles in the checklist tasks, then those eligibility profiles must exist in the target environment.

Related Topics

- Examples of Loading Checklist Templates

Examples of Loading Checklist Templates

This topic provides examples showing how to create checklist templates using HCM Data Loader. These examples reference all components using user keys.

Creating Enterprise Onboarding Checklists

This example Checklist.dat file creates:

- A parent enterprise onboarding checklist template. It has:
  - A child checklist template
  - A single task
  - A Contents component
- A child enterprise onboarding checklist template. It has:
  - Two tasks, which must be performed in the specified order
  - A notification override for one task
  - A Contacts component

```
COMMENT Create child and parent checklist templates
METADATA|Checklist|ChecklistName|Country|ChecklistCategory|AllocatedOn|CompletedOn|OffsetDays
MERGE|Checklist|Checklist_Child_1|United States|ORA_ONB_ENT_ONBOARDING_STEP|ORA_CHK_ALLOCATION_DT|
ORA_CHK_MAND_TASK_COMPLETE|0
MERGE|Checklist|Checklist_Parent_1|United States|ORA_ONB_ENT_ONBOARDING|ORA_CHK_ALLOCATION_DT|
ORA_CHK_MAND_TASK_COMPLETE|0

COMMENT Create checklist tasks for the parent and child checklist templates
METADATA|ChecklistTask|ChecklistName|Country|ChecklistCategory|ChecklistTaskName|MandatoryFlag|
DetailChecklistName|DetailChecklistCategory|DetailChecklistCountry|ActionType|PerformerRole|OwnerRole
MERGE|ChecklistTask|Checklist_Parent_1|United States|ORA_ONB_ENT_ONBOARDING|Task_Parent_1|Y|
Checklist_Child_1|ORA_ONB_ENT_ONBOARDING_STEP|United States|ORA_CHK_CHECKLIST|ORA_WORKER|ORA_CHK_INITIATOR
MERGE|ChecklistTask|Checklist_Child_1|United States|ORA_ONB_ENT_ONBOARDING_STEP|Task_Child_1|Y|||||
ORA_WORKER|ORA_CHK_INITIATOR
MERGE|ChecklistTask|Checklist_Child_1|United States|ORA_ONB_ENT_ONBOARDING_STEP|Task_Child_2|Y|||||
ORA_WORKER|ORA_CHK_INITIATOR

COMMENT Create the checklist task dependencies
METADATA|ChecklistTaskDependencies|DependencyType|ChecklistTaskName|ChecklistName|ChecklistCategory|Country|
DepenChecklistTaskName
MERGE|ChecklistTaskDependencies|ORA_DEPENDS_ON|Task_Child_2|Checklist_Child_1|ORA_ONB_ENT_ONBOARDING_STEP|
United States|Task_Child_1

COMMENT Create a checklist task notification override
METADATA|ChecklistTaskNotifications|NotifyOwner|NotifyPerformer|TaskEvent|ChecklistTaskName|ChecklistName|
ChecklistCategory|Country
MERGE|ChecklistTaskNotifications|Y|Y|ORA_CHK_TASK_ASSIGN|Task_Child_1|Checklist_Child_1|
ORA_ONB_ENT_ONBOARDING_STEP|United States

COMMENT Create a checklist contact
METADATA|ChecklistContacts|ContactTitleCode|ContactType|ChecklistName|ChecklistCategory|Country
MERGE|ChecklistContacts|ORA_ONB_ONBOARDING_SPONSOR|ORA_ONB_LN_MGR|Checklist_Child_1|
ORA_ONB_ENT_ONBOARDING_STEP|United States
```

```
COMMENT Create checklist contents
METADATA|ChecklistContents|ContentType|ChecklistName|Country|ChecklistCategory|ContentDefnCode
MERGE|ChecklistContents|ORA_ONB_EVENT|Checklist_Parent_1|United States|ORA_ONB_ENT_ONBOARDING|
CHKLIST_CONT_DEF67
```

This example ChecklistContentDetails.dat file creates the content details for the Contents component of the parent checklist template. This should be loaded before loading Enterprise Onboarding checklists.

```
COMMENT Create checklist content details
METADATA|ChecklistContentDetails|ContentCategory|ContentDefnCode|ContentSubtype|ContentType|Title|Status
MERGE|ChecklistContentDetails|ORA_ONB_INSIGHT|CHKLIST_CONT_DEF67|ORA_ONB_EVENT_STANDARD|ORA_ONB_EVENT|
Recruitment Drive|A
```

Creating Other Category Checklists

This example Checklist.dat file creates a checklist template of other category. It has:

- Two tasks, which must be performed in the specified order
- A notification override for one of the tasks

```
COMMENT Create the checklist template
METADATA|Checklist|ChecklistName|Country|ChecklistCategory|AllocatedOn|CompletedOn|OffsetDays
MERGE|Checklist|Offboarding Checklist|United States|OFFBOARD|ORA_CHK_ALLOCATION_DT|
ORA_CHK_MAND_TASK_COMPLETE|0

COMMENT Create the checklist tasks
METADATA|ChecklistTask|ChecklistName|Country|ChecklistCategory|ChecklistTaskName|MandatoryFlag|
PerformerRole|OwnerRole
MERGE|ChecklistTask|Offboarding Checklist|United States|OFFBOARD|Return laptop|Y|ORA_WORKER|
ORA_CHK_INITIATOR
MERGE|ChecklistTask|Offboarding Checklist|United States|OFFBOARD|Complete Exit Interview|Y|ORA_WORKER|
ORA_CHK_INITIATOR

COMMENT Create the checklist task dependencies
METADATA|ChecklistTaskDependencies|DependencyType|ChecklistTaskName|ChecklistName|ChecklistCategory|Country|
DepenChecklistTaskName
MERGE|ChecklistTaskDependencies|ORA_DEPENDS_ON|Complete Exit Interview|Offboarding Checklist|OFFBOARD|United
 States|Return laptop

COMMENT Create a task notification override
METADATA|ChecklistTaskNotifications|NotifyOwner|NotifyPerformer|TaskEvent|ChecklistTaskName|ChecklistName|
ChecklistCategory|Country
MERGE|ChecklistTaskNotifications|Y|Y|ORA_CHK_TASK_ASSIGN|Return laptop|Offboarding Checklist|ONBOARD|United
 States
```

Creating Checklists and Tasks with Display Properties

This example Checklist.dat file creates the display properties for checklists and tasks.

```
METADATA|Checklist|ChecklistName|ChecklistCategory|ActiveInactiveFlag|Description|Country
MERGE|Checklist|Onboarding Checklist|ONBOARD|ORA_CHK_ACTIVE|Checklist for Onboarding New Hires|
METADATA|ChecklistTask|ChecklistName|ChecklistCategory|Country|ChecklistTaskName|MandatoryFlag|
ActiveInactiveFlag|PerformerRole|ActionType
MERGE|ChecklistTask|Onboarding Checklist|ONBOARD||Collect ID card|N|ORA_CHK_TASK_ACTIVE|ORA_WORKER|
ORA_CHK_MANUAL
METADATA|ChecklistProperty|ChecklistName|ChecklistCategory|Property|ActionType|ActionSubType|Manager|
Assignee|Other|Performer|Owner|PropertyValue
MERGE|ChecklistProperty|Onboarding Checklist|ONBOARD|ORA_COMPLETED_TASKS|||Hide|Hide|Hide|||
MERGE|ChecklistProperty|Onboarding Checklist|ONBOARD|ORA_OTHER_INCOMPLETE_TASKS|||H|H|H|||
MERGE|ChecklistProperty|Onboarding Checklist|ONBOARD|ORA_ACTION_REMOVE|ORA_CHK_EXTERNAL_URL||H||Hide|Show|V|
MERGE|ChecklistProperty|Onboarding Checklist|ONBOARD|ORA_ACTION_REMOVE|ORA_CHK_ESIGN|ORA_BASIC_SIGN|H||Hide|
Show|V|
```

```
METADATA|ChecklistTaskProperty|ChecklistName|ChecklistCategory|Property|ChecklistTaskName|Manager|Other|
Performer|Owner
MERGE|ChecklistTaskProperty|Onboarding Checklist|ONBOARD|ORA_ACTION_REASSIGN|Collect ID card|Show|Show|Show|
Show
MERGE|ChecklistTaskProperty|Onboarding Checklist|ONBOARD|ORA_ACTION_EDIT_DUE_DATE|Collect ID card|V|V|Hide|H
```

This example Checklist.dat file creates the combine task notifications setting for checklists.

```
METADATA|Checklist|ChecklistName|ChecklistCategory|ActiveInactiveFlag|Description|Country
MERGE|Checklist|Onboarding Checklist|ONBOARD|ORA_CHK_ACTIVE|Checklist for Onboarding New Hires|
METADATA|ChecklistProperty|ChecklistName|ChecklistCategory|Property|ActionType|ActionSubType|Manager|
Assignee|Other|Performer|Owner|PropertyValue
MERGE|ChecklistProperty|Onboarding Checklist|ONBOARD|ORA_COMBINE_TASK_NOTIFICATIONS|||||||Yes
```

Adding Tasks to a Checklist Template

This example Checklist.dat file adds tasks to a checklist template.

```
METADATA|ChecklistTask|ChecklistName|Country|ChecklistCategory|ChecklistTaskName|Description|MandatoryFlag|
ActionType
MERGE|ChecklistTask|Offboarding Checklist|United States|OFFBOARD|Return ID card|Return ID card to security|
Y|Manual task
```

Updating Checklists and Tasks

This example Checklist.dat file updates a checklist template and its tasks. This is for other checklist categories.

```
METADATA|Checklist|ChecklistName|Country|ChecklistCategory|Description|ActiveInactiveFlag|AllocatedOn|
CompletedOn
MERGE|Checklist|Offboarding Checklist|United States|OFFBOARD|Checklist Creation 1|ORA_CHK_ACTIVE|
ORA_CHK_ALLOCATION_DT|ORA_CHK_MAND_TASK_COMPLETE
METADATA|ChecklistTask|ChecklistName|Country|ChecklistCategory|ChecklistTaskName|Description|MandatoryFlag|
ActionURL|ActionType
MERGE|ChecklistTask|Offboarding Checklist|United States|OFFBOARD|Submit Expenses|Submit Expense Report|Y|
www.oracle.com|External URL
```

Deleting a Checklist and Associated Objects

This example Checklist.dat file deletes a checklist template and all its associated objects. This is for other checklist categories.

```
METADATA|Checklist|ChecklistName|Country|ChecklistCategory
DELETE|Checklist|Offboarding Checklist|United States|OFFBOARD
```

Creating Tasks in the Task Library

This example TaskLibrary.dat file creates checklist tasks in the task library.

```
METADATA|TaskLibrary|TaskCode|TaskName|Description|MandatoryFlag|ActiveInactiveFlag|ActionType|ActionUrl|
TaskActionCode|PerformerRole|PerformerRespType|OwnerRole
MERGE|TaskLibrary|APPLY_FOR_CREDIT_CARD|Apply for Credit Card||Y|ORA_CHK_TASK_ACTIVE|ORA_CHK_MANUAL|||
ORA_RESP_TYPE|HR_REP|ORA_CHK_INITIATOR
MERGE|TaskLibrary|VERIFY_PERSONAL_DETAILS|Verify personal details|Application Task to verify personal
 details|N|ORA_CHK_TASK_ACTIVE|ORA_CHK_APP_TASK||Me.me_personal_details|||
MERGE|TaskLibrary|HOW_TO_ONBOARD_PPL|External URL - How to onboard people||N|ORA_CHK_TASK_ACTIVE|
ORA_CHK_EXTERNAL_URL|https://docs.oracle.com/en/cloud/saas/human-resources/20c/fawhr/hire-and-manage-
workers.html#FAWHR3025069||||
MERGE|TaskLibrary|WELCOME_TO_ORACLE|Video - Welcome to Oracle||Y|ORA_CHK_TASK_ACTIVE|ORA_CHK_VIDEO|https://
youtube.oracle.com/embed/klc57AoU6eY||||
```

This example TaskLibrary.dat file creates the display properties for checklist tasks in the task library.

```
METADATA|TaskLibrary|TaskCode|TaskName|Description|MandatoryFlag|ActiveInactiveFlag|ActionType
MERGE|TaskLibrary|APPLY_FOR_CREDIT_CARD|Apply for Credit Card||Y|ORA_CHK_TASK_ACTIVE|ORA_CHK_MANUAL
METADATA|TaskLibraryProperty|TaskCode|Property|Manager|Other|Performer|Owner
MERGE|TaskLibraryProperty|TASK_LIB_HDL|ORA_ACTION_REASSIGN|V|H|V|V
MERGE|TaskLibraryProperty|TASK_LIB_HDL|ORA_ACTION_EDIT_DUE_DATE|V|Hide|H|H
```

Related Topics

- Guidelines for Loading Checklist Templates
- Considerations When Using Enterprise Onboarding Journey Category

Display Properties for Checklists and Tasks

When you create a checklist, you can selectively display only those sections or actions that you want a role to access. You can do this by configuring the display of the checklist or task property for that role.

For example, in an off boarding checklist, you may want to restrict line managers from seeing tasks that other users need to be complete. You can hide the section that lists others' incomplete tasks only for line managers. Similarly, you can configure the display at the task level.

This topic lists the display values you can configure at the checklist and task level for a role. If you don't make a change, the default is used.

Checklist Display Properties

Checklist display property	Code	Display values for assignee	Display values for line manager	Display values for other user
What's Happening	ORA_ANNOUNCEMENTS	• Show (default) • Hide	• Not Applicable	• Not Applicable
Basic Info	ORA_BASIC_INFORMATION	• Not Applicable	• Show (default) • Hide	• Show (default) • Hide
Completed Tasks on Employee Progress Page	ORA_COMPLETED_TASKS	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide
Contact Us	ORA_CONTACT_US	• Not Applicable	• Show (default) • Hide	• Show (default) • Hide
Deferred Tasks	ORA_DEFERRED_TASKS	• Not Applicable	• Show (default) • Hide	• Show (default) • Hide
Fun Stuff	ORA_FUN_STUFF	• Show (default) • Hide	• Not Applicable	• Not Applicable
Note board	ORA_NOTE_BOARD	• Show (default) • Hide	• Not Applicable	• Not Applicable

Checklist display property	Code	Display values for assignee	Display values for line manager	Display values for other user
Completed Tasks on Employee Onboarding Page	ORA_ONB_COMPLETED_TASKS	• Show (default) • Hide	• Not Applicable	• Not Applicable
Order Tasks by Sequence	ORA_ORDER_BY_SEQUENCE	• No (default) • Yes	• No (default) • Yes	• No (default) • Yes
Others' Incomplete Tasks	ORA_OTHER_INCOMPLETE_TASKS	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide
Tasks to Finish	ORA_THINGS_TO_FINISH	• Show (default) • Hide	• Not Applicable	• Not Applicable
Add a Task from Task Library	ORA_ACTION_ADD_LIBRARY_TASK	• Not Applicable	• Show (default) • Hide*	• Show (default) • Hide*
Add a Task	ORA_ACTION_ADD_TASK	• Not Applicable	• Show (default) • Hide**	• Show (default) • Hide**
Edit Checklist	ORA_ACTION_EDIT_CHECKLIST	• Not Applicable	• Show (default) • Hide	• Show (default) • Hide
Force Complete a Checklist	ORA_ACTION_FORCE_CLOSE	• Not Applicable	• Show (default) • Hide	• Show (default) • Hide
Remove a Checklist	ORA_ACTION_REMOVE	• Not Applicable	• Show (default) • Hide	• Show (default) • Hide

- *Based on the security privilege PER_ADD_TASK_FOR_WORKER_FROM_TASK_LIBRARY_PRIV
- **Based on the security privilege PER_ADD_NEW_TASK_FOR_WORKER_PRIV

Task Display Properties

Using task display properties you can control the access for the assignee, owner, line manager, and other users. Whatever you configure at the task level overrides what's configured at the checklist and task type level. If you don't make a change, the default is used.

Task display property	Code	Display values for assignee	Display values for owner	Display values for line manager	Display values for other user
Task Access	ORA_DISPLAY_ACTIVITY	• Allow (default) • Show • Hide	• Allow (default) • Show • Hide	• Allow (default) • Show • Hide	• Allow (default) • Show • Hide
Attachments	ORA_DISPLAY_ATTACHMENTS	• Allow (default)	• Allow (default)	• Allow (default)	• Allow (default)

Task display property	Code	Display values for assignee	Display values for owner	Display values for line manager	Display values for other user
		• Show • Hide	• Show • Hide	• Show • Hide	• Show • Hide
Comments	ORA_DISPLAY_COMMENTS	• Allow (default) • Show • Hide	• Allow (default) • Show • Hide	• Allow (default) • Show • Hide	• Allow (default) • Show • Hide
Contact Information	ORA_DISPLAY_CONTACT	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide
Notes	ORA_DISPLAY_NOTES	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide
Mark a Task as Complete	ORA_ACTION_COMPLETE	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide
Edit a Task Due Date	ORA_ACTION_EDIT_DUE_DATE	• Show • Hide (default)	• Show (default) • Hide	• Show • Hide (default)	• Show (default) • Hide
Edit a Task	ORA_ACTION_EDIT_TASK	• Show • Hide (default)	• Show (default) • Hide	• Show • Hide (default)	• Show (default) • Hide
Reassign a Task	ORA_ACTION_REASSIGN	• Show • Hide (default)	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide
Mark a Task as Not Applicable	ORA_ACTION_REJECT	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide
Remove a Task	ORA_ACTION_REMOVE	• Show • Hide (default)	• Show (default) • Hide	• Show • Hide (default)	• Show (default) • Hide
Reopen a Task	ORA_ACTION_REOPEN	• Not Applicable	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide
Send Reminder	ORA_ACTION_SEND_REMINDER	• Not Applicable	• Show (default) • Hide	• Show (default) • Hide	• Show (default) • Hide

Task Action Labels

This table shows the action labels and code names for different task types:

Task Type	Action Label	Code Name
All Task Types	• Done	COMPLETE_ACTION_LABEL

Task Type	Action Label	Code Name
	• Complete	
All Task Types	Mark as Not Applicable	REJECT_ACTION_LABEL
All Task Types	Add To Calendar	CALENDAR_ACTION_LABEL
All Task Types	Save as Draft	SAVE_ACTION_LABEL
Advanced Task	Go to website	ACTIVITY_ACTION1_LABEL
Advanced Task	Go to application task	ACTIVITY_ACTION2_LABEL
Advanced Task	View document	ACTIVITY_ACTION3_LABEL
Electronic Signature - DocuSign	Sign Here	ACTIVITY_ACTION1_LABEL
External URL	• Go To Task • Go To Website	ACTIVITY_ACTION1_LABEL
I9 Verification - Section 1	Go To Task	ACTIVITY_ACTION1_LABEL
I9 Verification - Section 2	Go To Task	ACTIVITY_ACTION1_LABEL
I9 Verification - Verification Status	See Status	ACTIVITY_ACTION1_LABEL
I9 Verification - Verification Status	Upload to DOR	ACTIVITY_ACTION2_LABEL
Process Automation - Initiate	Initiate	ACTIVITY_ACTION1_LABEL
Process Automation - Verify Status	See Status	ACTIVITY_ACTION1_LABEL
Report	View Document	ACTIVITY_ACTION1_LABEL

Related Topics

- Guidelines for Loading Checklist Templates
- Examples of Loading Checklist Templates
- Examples of Configuring Journey and Task Display Properties
- Journey Display Properties

Examples of Loading Checklist and Task Display Properties

This topic looks at how you use display properties to restrict access both at the checklist and task level using HCM Data Loader.

Creating the Display Properties at the Checklist Level

In an off boarding checklist, you want to restrict line managers from seeing tasks that other users need to complete. You hide the Others' Incomplete Tasks section that lists others' incomplete tasks only for line managers. This example Checklist.dat file creates the display properties for a checklist section.

```
METADATA|Checklist|ChecklistName|ChecklistCategory|ActiveInactiveFlag|Description|Country
MERGE|Checklist|Offboarding Checklist|OFFBOARD|ORA_CHK_ACTIVE|Checklist for Offboarding|METADATA|
ChecklistTask|ChecklistName|ChecklistCategory|Country|ChecklistTaskName|MandatoryFlag|ActiveInactiveFlag|
PerformerRole|ActionType
MERGE|ChecklistTask|Offboarding Checklist|OFFBOARD||Submit Exit Survey|N|ORA_CHK_TASK_ACTIVE|ORA_WORKER|
ORA_CHK_MANUALMETADATA|ChecklistProperty|ChecklistName|ChecklistCategory|Property|ActionType|ActionSubType|
Manager|Assignee|Other|Performer|Owner|PropertyValue
MERGE|ChecklistProperty|Offboarding Checklist|OFFBOARD|ORA_OTHER_INCOMPLETE_TASKS|||Hide|||||
```

Creating the Display Properties at the Task Level

You configure a task of the type questionnaire to collect feedback from employees on their onboarding experience. You don't want this feedback to be visible to the line manager and other users. Therefore, you don't give line managers and other users access to the Task Access section. This example Checklist.dat file creates the display properties for a task section.

```
METADATA|Checklist|ChecklistName|ChecklistCategory|ActiveInactiveFlag|Description|Country
MERGE|Checklist|Onboarding Checklist|ONBOARD|ORA_CHK_ACTIVE|Checklist for Onboarding New Hires|METADATA|
ChecklistTask|ChecklistName|ChecklistCategory|Country|ChecklistTaskName|MandatoryFlag|ActiveInactiveFlag|
PerformerRole|ActionType|QuestionnaireCode
MERGE|ChecklistTask|Onboarding Checklist|ONBOARD||Onboarding Feedback|N|ORA_CHK_TASK_ACTIVE|ORA_WORKER|
ORA_CHK_QUESTIONNAIRE|ONBOARDING_SURVEYMETADATA|ChecklistTaskProperty|ChecklistName|ChecklistCategory|
Property|ChecklistTaskName|Manager|Other|Performer|Owner
MERGE|ChecklistTaskProperty|Onboarding Checklist|ONBOARD|ORA_DISPLAY_ACTIVITY|Onboarding Feedback|Hide|
Hide||
```

Overriding the Display Properties at the Task Level

You can override the display properties you specified at the checklist level at the task level. This example Checklist.dat file overrides the display properties for a task action.

```
METADATA|Checklist|ChecklistName|ChecklistCategory|Country|ActiveInactiveFlag|Description
MERGE|Checklist|Checklist Task Display|ONBOARD||ORA_CHK_ACTIVE|Demo task display property overrides at
 checklist level
METADATA|ChecklistTask|ChecklistName|ChecklistCategory|Country|ChecklistTaskName|MandatoryFlag|
ActiveInactiveFlag|ActionType|EsignType
MERGE|ChecklistTask|Checklist Task Display|ONBOARD||Manual Task|N|ORA_CHK_TASK_ACTIVE|ORA_CHK_MANUAL|
MERGE|ChecklistTask|Checklist Task Display|ONBOARD||Basic esign Task|N|ORA_CHK_TASK_ACTIVE|ORA_CHK_ESIGN|
ORA_BASIC_SIGN
METADATA|ChecklistProperty|ChecklistName|ChecklistCategory|Country|Property|ActionType|ActionSubType|
Manager|Other|Performer|Owner|PropertyValue

COMMENT ChecklistTaskProperty modification at checklist level - with ActionType
MERGE|ChecklistProperty|Checklist Task Display|ONBOARD||ORA_ACTION_REMOVE|ORA_CHK_MANUAL||Hide|Hide|Hide|
Hide|

COMMENT ChecklistTaskProperty modification at checklist level - with ActionType and ActionSubType
MERGE|ChecklistProperty|Checklist Task Display|ONBOARD||ORA_ACTION_REMOVE|ORA_CHK_ESIGN|ORA_BASIC_SIGN|Show|
Show|Hide|Show|
```

Related Topics

- Guidelines for Loading Checklist Templates
- Examples of Loading Checklist Templates
- Display Properties for Checklists and Tasks

Guidelines for Loading Document Delivery Preferences

Employers periodically deliver documents, such as payslips and year-end tax statements, to workers. Document delivery preferences specify how workers receive those documents.

For example, workers may receive their payslips online. You can specify a default delivery method for a document type, and you can override the default method on relevant work structures. For example, delivery preferences for payroll documents can be overridden at Payroll Statutory Unit level. You can also specify delivery preferences for a person. Preferences specified at person level override those at all other levels. This topic describes how to load Document Record Delivery Preference objects for a person using HCM Data Loader.

Document Type

Before you can load document delivery preferences for a person:

- The document type must exist in the target environment.
- Document delivery preferences must be enabled for the document type. In the document type definition, **Override Hierarchy** must be set to either **Payroll** or **General**, as appropriate. For example, if you're loading delivery preferences for performance documents, then **Override Hierarchy** must be set to **General**.

Loading Document Delivery Preferences

You load document delivery preferences for a person in a DocumentDeliveryPreference.dat file.

This example DocumentDeliveryPreference.dat file loads document delivery preference records. These records are for the document type identified by source key DT1345 and the three people identified by the person number user key.

```
METADATA|DocumentDeliveryPreference|DocumentTypeId(SourceSystemId)|LevelCode|PersonNumber|OnlineEnabledFlag|
PaperEnabledFlag|AllowWorkerOverrideFlag|OnlineConsentRequiredFlag|InitialConsentFlag|SourceSystemId|
SourceSystemOwner
MERGE|DocumentDeliveryPreference|DT1435|900_PERSON|Z8154257|Y|N|Y|Y|Y|LoadPref1|PSFT-US
MERGE|DocumentDeliveryPreference|DT1435|900_PERSON|Z8154806|N|Y|Y|Y|Y|LoadPref3|PSFT-US
MERGE|DocumentDeliveryPreference|DT1435|900_PERSON|Z8154813|Y|N|Y|Y|Y|LoadPref2|PSFT-US
```

Guidelines for Loading Document Types

A document type, such as Performance Improvement Plan or Letter of Recognition, defines the purpose and treatment of documents of that type. When defining a document type, you include attributes, such as the document name, dates, and issuing authority, and specify whether they're required.

You can also specify an expiration notification period, indicate whether approvals are required, and enable multiple occurrences of the document. This topic describes aspects of the Document Type object that you must understand to load document types successfully using HCM Data Loader.

Document Categories and Subcategories

Document types belong to a document category and may also belong to a document subcategory. Document categories, such as Expenses or Payroll, provide a way to group document types for ease of retrieval and management. Document subcategories, such as Involuntary Deduction or Additional Income in the Payroll category, provide a further level of detail.

Before loading document types, ensure that referenced document categories and subcategories exist in the target environment. In the Setup and Maintenance work area, use the following to create document categories:

- Functional Area: Workforce Information
- Task: Manage Document Lookups

In the Setup and Maintenance work area, use the following to create document subcategories:

- Functional Area: Workforce Information
- Task: Manage Extended Lookup Codes

Document Types

When loading document types, you must provide a unique reference for the record being created. If your document type isn't country-specific, then you must supply a source key. This rule exists because the **Country** attribute forms part of the user key.

If you're loading a Document Type Delivery Preference component, then set the **Hierarchy Code** attribute of the Document Type component to either **GENERAL** or **PAYROLL**. If you omit this attribute or provide no value, then document delivery preferences aren't enabled for the document type.

The YES_NO_REQUIRED lookup is used for the Relevant and Required attributes when configuring a document type in HCM Data Loader. This table describes the lookup details:

Code	Value	Meaning
Y	Yes	Relevant
N	No	Not Relevant
R	Required	Required

Document Type Delivery Preferences

The Document Type Delivery Preference component enables users to specify delivery preferences for documents of the associated type. For example, you could enable users to specify that payslips can be delivered both on paper and online.

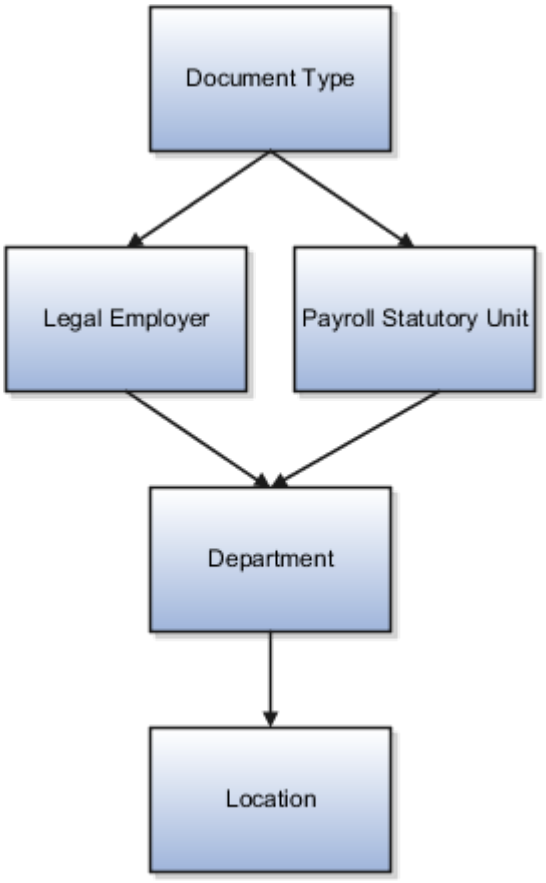
The delivery preferences for the document type can be overridden. That is, you can specify document delivery preferences:

- At the payroll statutory unit level, for documents in the PAYROLL category

- At the legal employer level, for documents in all other categories

In both cases, delivery preferences can be further overridden at department and location levels.

This figure summarizes the override hierarchy for document delivery preferences. Entries at any level of the hierarchy override those above them.



If you set the **Allow Worker Override** attribute to **Y**, then workers can specify document delivery preferences for themselves. Worker preferences override preferences at all other levels.

Related Topics

- Examples of Loading Document Types
- How You Set Preferences for Document Delivery
- Document Types and Categories

Examples of Loading Document Types

This topic provides examples showing how to load and manage Document Type objects using HCM Data Loader.

Creating Document Types Using Source Keys

This example DocumentType.dat file creates one global and one country-specific document type. The document types are identified using source keys.

```
METADATA|DocumentType|SourceSystemOwner|SourceSystemId|DocumentType|Description| LegislationCode|
CategoryCode|AuthorizationRequiredFlag|MultipleOccurencesFlag|ActiveInactiveFlag|PublishRequiredFlag|
HierarchyCode|DocumentNameRequired|DocumentNumberRequired|DateFromRequired|DateToRequired|
IssuingCountryRequired|IssuingLocationRequired|IssuingAuthorityRequired|IssuedDateRequired|CommentsRequired
MERGE|DocumentType|VISION|DTYPETest001|RS Global DocType1|RS Global DocType1 Description||PAYROLL|Y|Y|N|Y|
GENERAL|R|R|Y|Y|Y|Y|Y|Y
MERGE|DocumentType|VISION|DTYPETest003|RS US DocType1|RS US DocType1 Description|US|AUDIT|Y|Y|N|Y|PAYROLL|R|
R|Y|Y|Y|Y|Y|Y|Y
```

Creating Document Types Using User Keys

This example DocumentType.dat file creates one country-specific document type. The document type is identified using user keys.

```
METADATA|DocumentType|DocumentType|Description|Country|CategoryCode|AuthorizationRequiredFlag|
MultipleOccurencesFlag|ActiveInactiveFlag|PublishRequiredFlag|HierarchyCode|DocumentNameRequired|
DocumentNumberRequired|DateFromRequired|DateToRequired|IssuingCountryRequired|IssuingLocationRequired|
IssuingAuthorityRequired|IssuedDateRequired|CommentsRequired
MERGE|DocumentType|US Audit Doc Type|US Audit Doc Type Description|United States|AUDIT|Y|Y|Y|Y|PAYROLL|R|R|
Y|Y|Y|Y|Y|Y|Y
```

Note: If the document type were global, you would have to provide a source key rather than a user key. This requirement exists because **Country** is a required attribute of the user key.

Creating Document Types with Delivery Preferences

This example DocumentType.dat file creates a country-specific document type with delivery preferences for the document type, payroll statutory unit, and department. The document type and delivery preferences are identified using source keys.

```
METADATA|DocumentType|SourceSystemOwner|SourceSystemId|DocumentType|Description| LegislationCode|
CategoryCode|AuthorizationRequiredFlag|MultipleOccurencesFlag|ActiveInactiveFlag|PublishRequiredFlag|
HierarchyCode|DocumentNameRequired|DocumentNumberRequired|DateFromRequired|DateToRequired|
IssuingCountryRequired|IssuingLocationRequired|IssuingAuthorityRequired|IssuedDateRequired|CommentsRequired
MERGE|DocumentType|VISION|DTYPETest003|RS US DocType1|RS US DocType1 Description|US|AUDIT|Y|Y|N|Y|PAYROLL|R|
R|Y|Y|Y|Y|Y|Y|Y
METADATA|DeliveryPreference|SourceSystemOwner|SourceSystemId|DocumentTypeId(SourceSystemId)|
InitialConsentValueFlag|OnlineConsentRequiredFlag|OnlineEnabledFlag|PaperEnabledFlag|
AllowWorkerOverrideFlag|EmailEnabledFlag|LegislationCode|LevelCodeName|DocumentTypeCountry|
DocumentTypeLegislationCode|PayrollStatutoryUnitName|DepartmentName
MERGE|DeliveryPreference|VISION|DTYPETest003_Pref0|DTYPETest003|N|N|Y|Y|N|N|US||US|US||
MERGE|DeliveryPreference|VISION|DTYPETest003_Pref1|DTYPETest003|N|N|Y|Y|N|N|US|Payroll Statutory Unit|US|US|
GBI HCM Widgets USA|
MERGE|DeliveryPreference|VISION|DTYPETest003_Pref2|DTYPETest003|N|N|Y|Y|N|N|US|Department|US|US|GBI HCM
 Widgets USA|HCM-1001-Corporate
```

Loading Translated Document Types and Descriptions

This example DocumentTypeTranslation.dat file translates an existing document type and its description. It identifies the document type using source keys.

```
METADATA|DocumentTypeTranslation|SourceSystemOwner|SourceSystemId|BaseDocumentType|LegislationCode|
SourceLang|Language|DocumentType|Description
MERGE|DocumentTypeTranslation|VISION|DTYPETest001|RS Global DocType1||US|JP|RS Global DocType1 JP|RS Global
 DocType1 DESC JP
MERGE|DocumentTypeTranslation|VISION|DTYPETest003|RS US DocType1|US|US|JP|RS US DocType1 JP|RS US DocType1
 DESC JP
```

Deleting Document Types

You can delete a document type only if no document records of that type exist. When you delete a document type, any associated delivery preferences are also deleted. This example DocumentType.dat file deletes a document type that's identified by its source key.

```
METADATA|DocumentType|SourceSystemOwner|SourceSystemId
DELETE|DocumentType|VISION|DTYPETest001
DELETE|DocumentType|VISION|DTYPETest003
```

This example DocumentType.dat file deletes a document type that's identified by its user key.

```
METADATA|DocumentType|DocumentType|Description|Country
DELETE|DocumentType|US Audit Doc Type|US Audit Doc Type Description|United States
```

Related Topics

- Guidelines for Loading Document Types

Examples of Loading Extended Lookup Codes

You use extended lookup codes to provide subcategories for lookup codes. The associated lookup codes must already exist when you load Extended Lookup objects using HCM Data Loader.

If the lookup codes are for specific legislations, then you can supply extended lookup codes for those legislations only. For example, if a lookup code has the +FR tag, then you can supply extended lookup codes for the FR legislation code only. This topic provides examples showing how to load Extended Lookup objects using HCM Data Loader.

Creating Extended Lookup Codes

This example ExtendedLookupCode.dat file creates extended lookup codes for the CONTRACT_TYPE lookup type. It identifies the extended lookup codes using source keys.

```
METADATA|ExtendedLookupCode|SourceSystemOwner|SourceSystemId|LookupType|LookupCode|LegislationCode|
ExtendedLookupCode|ExtendedLookupCodeName
MERGE|ExtendedLookupCode|VISION|ELC_CONTRACT_LIMITED|CONTRACT_TYPE|5|NO|L|Limited Contract
MERGE|ExtendedLookupCode|VISION|ELC_CONTRACT_DIRECTOR|CONTRACT_TYPE|FR_CEO_MANDATE|FR|DC|Director Contract
```

Loading Translated Extended Lookup Code Names

Supply the names of extended lookup codes in the language of the user who's loading them. You supply an ExtendedLookupCodeTranslation.dat file to translate the names of extended lookup codes into other enabled languages once the codes exist. This example translates the name of an existing extended lookup code. It identifies the code by its source key.

```
METADATA|ExtendedLookupCodeTranslation|SourceSystemOwner|SourceSystemId|Language|ExtendedLookupCodeName
MERGE|ExtendedLookupCodeTranslation|VISION|ELC_CONTRACT_LIMITED|FR|Contrat a Duree Limitee
```

Deleting Extended Lookup Codes

You can delete Extended Lookup objects using HCM Data Loader. This example ExtendedLookupCode.dat file deletes specific extended lookup codes for the CONTRACT_TYPE lookup code. It identifies the extended lookup codes using source keys.

```
METADATA|ExtendedLookupCode|SourceSystemOwner|SourceSystemId
DELETE|ExtendedLookupCode|VISION|ELC_CONTRACT_LIMITED
DELETE|ExtendedLookupCode|VISION|ELC_CONTRACT_DIRECTOR
```

Guidelines for Loading Name Formats

A name format is a set of rules for combining individual name components, such as first name and last name, to form a complete person name. Name formats are specific to a legislation and name-format type.

The name-format types are display name, list name, order name, and full name. The predefined global format is used when no format exists for a format type and legislation. This topic describes how to load Name Format objects using HCM Data Loader.

Understanding Format Masks

A format mask is a string of codes used to construct a name format. The codes represent the required name components, symbols, and special characters. Use the codes shown in this table to identify the name components.

Code	Name Component
FIR	First Name
LAS	Last Name
MID	Middle Name
PLN	Previous Last Name
KNA	Known As
HNS	Honors
PNA	Prefix
SUF	Suffix
TIT	Title
MLR	Military Rank
$INF1$ through **$INF30$**	Name Information 1 through Name Information 30

Use the codes shown in this table for symbols and special characters.

Code	Description	Value
SPA	space	space character ( )
COM	comma	,
OPE	left parenthesis	(
CLO	right parenthesis	)
QUO	quotation mark	"
DOT	period	.
SLA	slash	/
COL	colon	:
SEM	semicolon	;
ATT	at sign	@

Constructing a Format Mask

The format mask must:

- Start and end with a vertical bar (|).
- Separate each name component with two vertical bars (||).

The vertical bar is the default delimiter for HCM Data Loader .dat files. If you haven't selected a different default delimiter, then you must prefix the vertical bar in the name-format mask with the HCM Data Loader escape character. The escape character ensures that HCM Data Loader ignores delimiters in the format mask. The default escape character is the backslash (\). For example, to provide a format mask for the name format `Title Last Name, First Name (Known As)`, you supply the codes for each element of the name as shown in this table.

Element	Codes
Title space	$TIT$$SPA$
Last Name, space	$LAS$$COM$$SPA$
First Name space	$FIR$$SPA$
(Known As)	$OPE$$KNA$$CLO$

In the format mask, each name component must be delimited by two vertical bars:

```
$TIT$$SPA$\|\|$LAS$$COM$$SPA$\|\|$FIR$$SPA$\|\|$OPE$$KNA$$CLO$
```

In addition, the format mask must start and end with a single vertical bar:

```
\|$TIT$$SPA$\|\|$LAS$$COM$$SPA$\|\|$FIR$$SPA$\|\|$OPE$$KNA$$CLO$\|
```

Creating Name Formats

This example NameFormat.dat file creates a display name for France in the format `Title First Name Last Name`. The name format is identified by its source key.

```
METADATA|NameFormat|SourceSystemOwner|SourceSystemId|FormatName|LegislationCode|UserFormatChoice|FormatMask
MERGE|NameFormat|VISION|NF_FR_L_DISP|DISPLAY_NAME|FR|L|\|$TIT$$SPA$\|\|$FIR$$SPA$\|\|$LAS$\|
```

Deleting Name Formats

You can delete a Name Format object using HCM Data Loader, provided that the name format isn't being used. This example NameFormat.dat file deletes a name format. It identifies the name format by its source key.

```
METADATA|NameFormat|SourceSystemOwner|SourceSystemId
DELETE|NameFormat|VISION|NF_FR_L_DISP
```

Related Topics

- Person Name Formats

Examples of Loading Person Types

System person types are predefined values that identify groups such as employees and contingent workers. You can't create, edit, or delete system person types. However, each system person type is associated with one or more user person types, which further categorize the group.

You can create, edit, and delete user person types. For example, you could define Associate and Remote Worker as user person types of the Employee system person type. For any system person type, one user person type must be identified as the default value. This topic provides examples showing how to load and manage user Person Type objects using HCM Data Loader.

Creating Person Types

This example PersonType.dat file creates the Officer and Rating user person types for the Employee system person type. It identifies the person types using source keys.

```
METADATA|PersonType|SourceSystemOwner|SourceSystemId|SystemPersonType|UserPersonType|ActiveFlag|DefaultFlag
MERGE|PersonType|VISION|PT_EMP_OFFICER|EMP|Officer|Y|N
MERGE|PersonType|VISION|PT_EMP_RATING|EMP|Rating|Y|N
```

Loading Translated Person-Type Names

Supply person-type names in the language of the user who's loading them. You supply a PersonTypeTranslation.dat file to translate person-type names into other enabled languages once the person types exist. This example translates the name of an existing person type. The person type is identified by its source key.

```
METADATA|PersonTypeTranslation|SourceSystemOwner|SourceSystemId|Language|UserPersonType
MERGE|PersonTypeTranslation|VISION|PT_EMP_OFFICER|FR|Officier
```

Deleting Person Types

You can delete Person Type objects using HCM Data Loader, provided that the person type isn't being used. This example PersonType.dat file deletes person types and any translated versions of the person-type names. It identifies the person types by their source keys.

```
METADATA|PersonType|SourceSystemOwner|SourceSystemId
DELETE|PersonType|VISION|PT_EMP_OFFICER
DELETE|PersonType|VISION|PT_EMP_RATING
```

Examples of Loading Resource Exceptions

A resource exception is a deviation in availability from a work schedule or schedule assignment. The resource exception defines when a resource is unavailable. For example, a worker may be attending training and therefore unavailable between specified dates.

You create resource exceptions, which are optional, for specific work schedules or schedule assignments. The work schedule or schedule assignment must exist before you can create an associated resource exception. This topic provides examples showing how to load Resource Exception objects using HCM Data Loader.

Creating Resource Exceptions

This example ResourceException.dat file creates a resource exception for a hospital appointment. It identifies the resource exception by its source key.

```
METADATA|ResourceException|SourceSystemOwner|SourceSystemId|ExceptionName|StartDateTime|EndDateTime
MERGE|ResourceException|VISION|RE_VISION_HOSPITAL|Hospital Appointment|2015/08/15 08:00:00|2015/08/15
 17:00:00
```

Deleting Resource Exceptions

You can delete a Resource Exception object using HCM Data Loader, provided that it's not referred to by a work schedule. This example ResourceException.dat file deletes an unused resource exception. It identifies the resource exception by its source key.

```
METADATA|ResourceException|SourceSystemOwner|SourceSystemId
DELETE|ResourceException|VISION|RE_VISION_HOSPITAL
```

Guidelines for Loading Time and Labor Event Groups

An event group is a group of related events, such as changes to a worker's assignment, that require a retroactive recalculation of time cards. This topic explains some aspects of event groups that you must understand to load them successfully using HCM Data Loader.

Loading Event Groups

These rules apply to the Event Group component:

- You can set **EventGroupCode** and **EventGroupName** to any user-defined value that identifies the purpose of the group.
- To resubmit time cards, you must set **EventGroupType** to **A** (Action).

These rules apply to the Date Tracked Event, Event Value Change, and Event Value Qualifier components:

- If you set **UpdateType** to **DT_INSERT**, then leave **ColumnName** blank.
- **EventGroupCode** identifies the group to which the component belongs.

These rules apply to the Event Value Change Component:

- **Sequence** specifies the sequence in which qualifying conditions are loaded. The sequence is important, as conditions can exist in a hierarchical relationship to each other. The sequence number also provides a way for other nodes in the condition hierarchy to refer to a condition.
- **ValidEvent** must be **Y** or **N**. It specifies whether the condition qualifies or disqualifies the event. For example, your qualifying condition may specify that an event is valid for any location change, except when moving from New York to San Francisco. In this case, you define two rows as shown in this table:

Condition	From Value	To Value	Valid Event
Parent	Any	Any	Y
Child	New York	San Francisco	N

- **FromValue** and **ToValue** can be any valid value for the column or **<ANY VALUE>** to indicate that any value can trigger resubmission of time cards.
- **ParentEvtValChangeSequence** is the sequence number of the parent Event Value Change component in the condition hierarchy.

These rules apply to the Event Value Qualifier component:

- **Sequence** is the sequence of the topmost Event Value Change component in the value change hierarchy for given Date Tracked Event and Event Value Qualifier components.
- **QualifierName** is the qualifier name, which is predefined in the PAY_EVENT_QUALIFIERS_F table.
- **QualifierValue** must be **Y** or **N**. If you set this value to **Y**, then updates to the worker's primary assignment trigger the event. If you set this value to **N**, then updates to the worker's secondary assignments trigger this event.

Related Topics

- Examples of Loading Time and Labor Event Groups
- Columns and Event Source Objects for Time and Labor Events

Examples of Loading Time and Labor Event Groups

An event group is a group of related events, such as changes to a worker's assignment, that require a retroactive recalculation of time cards. This topic shows how to create event groups using HCM Data Loader.

Loading Event Groups

This example EventGroup.dat file creates a single Event Group component, which is identified by its user key. For resubmitting time cards, the **EventGroupType** attribute value must be **A**.

```
METADATA|EventGroup|EventGroupCode|EventGroupName|EventGroupType
MERGE|EventGroup|JobChangeEventGroup|JobChangeEventGroup|A
```

Loading Date Tracked Events

The Date Tracked Event component identifies the type of event (update, correction, or insert) that triggers resubmission of time cards. It also identifies the affected object and the affected column or value in that object. This example identifies the Date Tracked Event component by its user key.

```
METADATA|DateTrackedEvent|ColumnName|UpdateType|EventGroupCode|DatedObjectName
MERGE|DateTrackedEvent|JOB_ID|DT_UPDATE_COLUMN|JobChangeEventGroup|
oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignmentDEO
```

Loading Event Value Changes

The Event Value Change component specifies the value changes that can cause time cards to be resubmitted. For example, the Date Tracked Event component may be monitoring changes to a worker's job. In that case, the Event Value Change component could specify the current and new jobs that trigger resubmission of time cards. This example identifies the Event Value Change component by its user key.

```
METADATA|EventValueChange|ColumnName|UpdateType|EventGroupCode|DatedObjectName|Sequence|ValidEvent|
FromValue|ToValue|EffectiveStartDate|EffectiveEndDate
MERGE|EventValueChange|JOB_ID|DT_UPDATE_COLUMN|JobChangeEventGroup|
oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignmentDEO|1|Y|<ANY_VALUE>|<ANY_VALUE>|
1950/01/01|4712/12/31
```

Loading Event Value Qualifiers

Using the optional Event Value Qualifier component, you can qualify an event. For example, when monitoring job changes, you could use this component to specify that only job changes in primary assignments are of interest. This example identifies the Event Value Qualifier component by its user key.

```
METADATA|EventValueQualifier|ColumnName|UpdateType|EventGroupCode|DatedObjectName|Sequence|
EffectiveStartDate|EffectiveEndDate|QualifierName|QualifierValue
MERGE|EventValueQualifier|JOB_ID|DT_UPDATE_COLUMN|JobChangeEventGroup|
oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignmentDEO|1|1950/01/01|4712/12/31|
EmployeeAssignmentDEO Primary Flag|Y
```

Related Topics

- Guidelines for Loading Time and Labor Event Groups
- Examples of Loading Time and Labor Event Actions
- Columns and Event Source Objects for Time and Labor Events

Columns and Event Source Objects for Time and Labor Events

When defining a Time and Labor event, you must provide values for the Column Name **(ColumnName)** and Event Source Object Name **(DatedObjectName)** attributes. This topic lists the values of these attributes for each Time and Labor event.

Event	Column Name	Event Source Object Name
Marital Status	`MARITAL_STATUS`	`oracle.apps.hcm.people.core.protectedModel.e`
Home Location	`ADDRESS_ID`	`oracle.apps.hcm.addresses.publicModel.entity`
Effective Start Date	`EFFECTIVE_START_DATE`	`oracle.apps.hcm.employment.core.publicModel.`
Effective End Date	`EFFECTIVE_END_DATE`	`oracle.apps.hcm.employment.core.publicModel.`
Business Unit	`BUSINESS_UNIT_ID`	`oracle.apps.hcm.employment.core.publicModel.`
Grade Ladder	`GRADE_LADDER_PGM_ID`	`oracle.apps.hcm.employment.core.publicModel.`
Working Hours	`NORMAL_HOURS`	`oracle.apps.hcm.employment.core.publicModel.`
Standard Working Hours	`TOTAL_HOURS`	`oracle.apps.hcm.employment.core.publicModel.`
Job	`JOB_ID`	`oracle.apps.hcm.employment.core.publicModel.`
Grade	`GRADE_ID`	`oracle.apps.hcm.employment.core.publicModel.`
Is Manager	`MANAGER_FLAG`	`oracle.apps.hcm.employment.core.publicModel.`
Location	`LOCATION_ID`	`oracle.apps.hcm.employment.core.publicModel.`
Position	`POSITION_ID`	`oracle.apps.hcm.employment.core.publicModel.`

Event	Column Name	Event Source Object Name
Worker Category	EMPLOYEE_CATEGORY	oracle.apps.hcm.employment.core.publicModel.
Assignment Category	EMPLOYMENT_CATEGORY	oracle.apps.hcm.employment.core.publicModel.
Regular or Temporary	PERMANENT_TEMPORARY_FLAG	oracle.apps.hcm.employment.core.publicModel.
Full-Time or Part-Time	FULL_PART_TIME	oracle.apps.hcm.employment.core.publicModel.
Person Type	PERSON_TYPE_ID	oracle.apps.hcm.employment.core.publicModel.
Hourly Paid or Salaried	HOURLY_SALARIED_CODE	oracle.apps.hcm.employment.core.publicModel.
Frequency	FREQUENCY	oracle.apps.hcm.employment.core.publicModel.
Working at Home	WORK_AT_HOME	oracle.apps.hcm.employment.core.publicModel.
Legal Employer	LEGAL_ENTITY_ID	oracle.apps.hcm.employment.core.publicModel.
Department	ORGANIZATION_ID	oracle.apps.hcm.employment.core.publicModel.
Reporting Establishment	ESTABLISHMENT_ID	oracle.apps.hcm.employment.core.publicModel.
Bargaining Unit	BARGAINING_UNIT_CODE	oracle.apps.hcm.employment.core.publicModel.
Collective Agreement	COLLECTIVE_AGREEMENT_ID	oracle.apps.hcm.employment.core.publicModel.
Union Member	LABOR_UNION_MEMBER_FLAG	oracle.apps.hcm.employment.core.publicModel.
Overtime Period	OVERTIME_PERIOD	oracle.apps.hcm.employment.core.publicModel.
Manager	MANAGER_ID	oracle.apps.hcm.employment.core.publicModel.
Work Day Definition	WORK_DAY_DEF_ID	oracle.apps.hcm.employment.core.publicModel.
Assignment DFFs	Provide the column name for the global context DFF.	oracle.apps.hcm.employment.core.publicModel.

Related Topics

- Examples of Loading Time and Labor Event Groups
- Guidelines for Loading Time and Labor Event Groups

Examples of Loading Time and Labor Event Actions

An event action defines how to react to an event, for example, whether to recalculate time cards when a worker's job or location changes. This topic shows how to create event actions using HCM Data Loader.

Loading Event Actions

This example EventAction.dat file creates a single Event Action component, which is identified by its user key.

```
METADATA|EventAction|EventActionCode|EffectiveStartDate|EffectiveEndDate|EventActionName|
EventActionTypeCode|LookbackTimeDefinitionCode|ActionSubmission|ProcessMode|AutoApprove
MERGE|EventAction|JobChangeAction|1950/01/01|4712/12/31|JobChangeAction|ORA_HWM_ACTION_TYPE_TC_RESUBMIT|
12_MONTHS_AGO_MONTH_START_DATE|SYNC|MANUAL|Y
```

These rules apply:

- To resubmit time cards, you must set **EventActionTypeCode** to **ORA_HWM_ACTION_TYPE_TC_RESUBMIT**.
- The **LookbackTimeDefinitionCode** value identifies a time period before the action execution date. Time cards from this period are considered for resubmission. Use one of these values:
    - 3_MONTHS_AGO_MONTH_START_DATE
    - 6_MONTHS_AGO_MONTH_START_DATE
    - 12_MONTHS_AGO_MONTH_START_DATE

  For example, a job change is effective on 1st January 2018 and the time-card resubmission process runs on that date. If **LookbackTimeDefinitionCode** is 3_MONTHS_AGO_MONTH_START_DATE, then time cards from 1st October 2017 are considered for resubmission.

  ActionSubmission must be **SYNC** for time-card resubmission.
- **ProcessMode** determines whether time cards are resubmitted automatically by the process or marked for resubmission but manually resubmitted. Set it to **AUTO** or **MANUAL**, as appropriate.
- **AutoApprove** determines whether time cards are approved automatically after they're resubmitted. Valid values are **Y** and **N**.

Loading Event Action Criteria

The Event Action Criteria component associates an event action with a specific group of workers. This Event Action Criteria component is identified by its user key.

```
METADATA|EventActionCriteria|EffectiveEndDate|EffectiveStartDate|GUID|SourceSystemId|SourceSystemOwner|
HCMGroupCode|EventActionCode|LegislativeDataGroupName
MERGE|EventActionCriteria|JobChangeAction|1950/01/01|4712/12/31|ABC_GRP
```

HcmGroupCode values are defined in the GRP_CODE column of the HWM_GRPS_VL table view and can be accessed using SQL. You can omit the Event Action Criteria component if the action applies to all workers.

You need the legislative data group name when you create an event action criteria that's referenced by an event action with a similar legislative data group.

Loading Event Group Actions

The Event Group Action component associates an event action with an event group. This Event Group Action component is identified by its user key.

```
METADATA|EventGroupAction|EventActionCode|EffectiveStartDate|EffectiveEndDate|EventGroupCode
MERGE|EventGroupAction|JobChangeAction|1950/01/01|4712/12/31|JobChangeEventGroup
```

Related Topics

- Examples of Loading Time and Labor Event Groups

3 Loading Workers

Overview of Loading Workers

The Worker object includes these person details: name, address, and picture, and these employment details: assignment, work relationship, contract, global transfer, tax reporting unit, working hours, and seniority dates and hours. This topic describes the considerations for loading workers using HCM Data Loader (HDL).

The Worker Hierarchy

Select the Worker object on the View Business Objects page to review its component hierarchy:

Using Source Keys

You can't update most components of the worker hierarchy if you supply only a user key. This restriction exists because the attribute that you want to change is the attribute that's used to identify the record. For example, in the Person Address component, the **AddressLine1** attribute is used both to identify the address to update and to supply the new value. Therefore, you're recommended always to supply source keys when creating workers and use them when updating worker records.

Multiple Instances of a Component

A person can have multiple instances of some components, such as Person Address, Person Phone, and Person Email. When you load multiple instances of a component for a person, you must:

- Identify one of the records as primary using the **PrimaryFlag** attribute of the component.
- Process the occurrences together and include the parent Worker component in the file. If the worker already exists in Oracle HCM Cloud, then you can include just the primary-key attributes of the worker.

The Employment Model

You must understand the employment model in the legal employer to which you're loading work relationships and assignments. In any legal employer, the **Employment Model** option can be set to one of these values:

- 2 Tier - Single Assignment
- 2 Tier - Multiple Assignment
- 2 Tier - Single Contract - Single Assignment
- 2 Tier - Multiple Contract - Single Assignment

Regardless of the employment model, an employment terms record is always created for an assignment. Therefore, you must include an Employment Terms component in the .dat file. When loading multiple work relationships or assignments for a person, you must identify which is primary using the **PrimaryFlag** attribute. A person must have only one primary work relationship at a time and only one primary assignment in each work relationship.

Defining Referenced Values

Many components of the Worker object include values, such as person type and assignment status, that must exist in the target environment. Perform the tasks shown in this table to define relevant values before you load data.

Task	Description
Manage Actions	Defines the actions used to classify changes to employment data
Manage Person Types	Defines subcategories of the predefined person types, such as Employee and Nonworker
Manage Assignment Status	Defines status values, such as active, inactive, or suspended, for assignments

In addition, you must have reviewed and updated lists of values, such as address types, phone types, ethnicity, and marital status, before loading workers. You may have performed this step during implementation. If you're synchronizing assignments from positions, then you must enable position synchronization before you load assignments.

Worker Termination

You terminate work relationships, not workers. When you terminate a work relationship using HCM Data Loader, its child components, such as Assignment components, are terminated automatically. Don't try to terminate other child components of the worker object, such as Person Name. The person record must continue to exist and be returned in search results, for example.

Loading the Person Attributes Descriptive Flexfield

The Person Attributes (PER_PERSONS_DFF) descriptive flexfield isn't date-effective, and a person can have only one context value at a time. If you load a new context value, then it overwrites any existing context value.

Related Topics

- Employment Model
- Source Keys
- How You Review Lists of Values
- Sources of Business-Object Information
- Position Synchronization

Guidelines for Preparing to Load Workers

Before you load Worker objects, you must decide:

- Whether you want user accounts to be created automatically for those workers.
- Whether user credentials are to be sent automatically to new users.

- Which roles are to be provisioned automatically to new users. User accounts created without at least one role are automatically suspended.

This topic describes each of these decisions.

Creating User Accounts Automatically

The enterprise option **User Account Creation** controls whether user accounts are created automatically for new workers, regardless of how those workers are created. If user accounts are created automatically in your enterprise, then you can prevent accounts being created for individual workers. To prevent account creation, include the **GeneratedUserAccountFlag** attribute of the User Information component and set it to **N**.

> **Note:** If **User Account Creation** is set to **None**, then you can't override it for individual workers.

You can supply a user name in the User Information component of the Worker object that you upload. Otherwise, user names are in the default format for the default user category, as specified on the Security Console. The default user-name format is the primary work email. Any changes that you make to the default format for a user category apply to all new users in the category, regardless of how they're created.

> **Tip:** New users are added to the default user category. You can move an existing user to a different user category by setting the **UserCategory** attribute of the User object.

Sending User Credentials to New Users

If you create user accounts automatically for uploaded workers, then you can notify the users automatically of their user names and passwords. Set the **SendCredentialsEmailFlag** attribute of the User Information component to **Y** for any worker who should receive this mail. **SendCredentialsEmailFlag** is set to **N** by default.

If you set **SendCredentialsEmailFlag** to **Y** for any worker, then you must ensure that a valid notification template is enabled for the default user category for this event. Two predefined templates exist:

- The **New Account Template** is for notifying the user.
- The **New Account Manager Template** is for notifying the user's manager.

You can also create notification templates. Notification templates are managed on the Security Console. Any changes made to notification templates apply to the user category.

Provisioning Roles to New Users

When a user account is created automatically for a worker, roles are provisioned automatically to the user as specified by current role-provisioning rules. Confirm that appropriate role mappings exist for users created by bulk upload.

> **Note:** All user accounts must have at least one role to remain active. If appropriate role mappings don't exist and you're not loading roles for manual assignment, then the new user account is immediately suspended. To avoid this automatic suspension of the account, define role mappings for workers before you load those workers.

Related Topics

- Role Mappings

Effective Start and End Dates for Worker Components

Many components of the Worker object are date effective. When creating workers, you must ensure that the effective dates of individual worker components don't conflict.

This topic suggests an approach to setting the earliest effective start date and last effective end date for each date-effective component. You don't have to follow this guidance. However, you must ensure that effective dates are aligned for the complete Worker object.

This table suggests how to set earliest effective start dates and last effective end dates for date-effective worker components.

Component	Earliest Effective Start Date	Last Effective End Date
Worker	Earliest start date of the worker.	The end of time.
Person Address	On or after the earliest start date of the worker.	The end of time.
Person Legislative Data	Earliest start date of the worker.	The end of time.
Person Name	Earliest start date of the worker.	The end of time.
Person Visa	On or after the earliest start date of the worker.	The end of time.
Person Contact Relationship	On or after the earliest start date of the worker.	The effective end date of the contact relationship.
Employment Terms	The first employment terms must have an effective start date equal to the start date of the corresponding work relationship. Subsequent employment terms can have an effective start date on or after the start date of the corresponding work relationship.	The end of time.
Assignment	The first assignment must have an effective start date equal to the earliest effective start date of any corresponding employment terms. Subsequent assignments can have an effective start date on or after the earliest effective start date of the corresponding employment terms. Further restrictions on the earliest effective start date may exist, depending on the employment model being used.	The end of time.
Assignment Extra Information	On or after the earliest effective start date of the corresponding assignment.	The end of time.

Component	Earliest Effective Start Date	Last Effective End Date
Assignment Grade Steps	On or after the earliest effective start date of the corresponding assignment.	The effective end date of the grade steps.
Assignment Manager	On or after the earliest effective start date of the corresponding assignment.	The effective end date of the manager relationship.
Assignment Work Measure	On or after the earliest effective start date of the corresponding assignment.	The end of time.
Contract	The earliest effective start date of the corresponding employment terms.	The end of time.
Worker Extra Information	On or after the earliest start date of the worker.	The end of time.

Guidelines for Deleting Worker Components

You can delete some components of the Worker object but not the entire worker. For information about deletion support for individual Worker components, see the component details on the View Business Objects page.

This topic provides additional information about deleting some of the Worker components using HCM Data Loader.

This table provides information about the circumstances in which you can delete some components of the Worker object.

Component	Deletion Supported	Notes
Assignment Grade Steps	Yes	If you set the **GradeId** attribute of an assignment to **#NULL**, then any associated Assignment Grade Steps component is deleted automatically. Therefore, if you include a DELETE instruction for the Assignment Grade Steps component in the same .dat file as the assignment, an error occurs.
Person Address	Yes	Subject to country-specific regulations, you can delete any address. When multiple mailing addresses exist, you must delete all nonprimary mailing addresses before you can delete the primary mailing address. Alternatively, identify a new primary mailing address before you delete the current primary mailing address.
Person Email	Yes	When only one email exists, you can delete it. When multiple emails exist, you must delete all nonprimary emails before you can delete the primary email. Alternatively, identify a new

Component	Deletion Supported	Notes
		primary email before you delete the current primary email.
Person Legislative Data	Yes	One legislative data record must exist for a worker. You can't delete the only record. You can delete additional records.
Person Phone	Yes	When only one phone exists, you can delete it. When multiple phones exist, you must delete all nonprimary phones before you can delete the primary phone. Alternatively, identify a new primary phone before you delete the current primary phone.
User Information	No	The User Information component is available only when you create workers. This component isn't available when you update workers.
Person User Manual Roles	No	The Person User Manual Roles component is available only when you create workers. This component isn't available when you update workers. To remove a single role from a worker, you use the User object rather than the Worker object. To remove multiple roles from a worker, you use the User Role component of the User object.
Work Relationship	Yes	When deleting a work relationship, you must include the **CancelWorkRelationshipFlag** attribute with a value of **Y**.

How You Check for Duplicate Person Records

When you load person records using HCM Data Loader, you can request a check for duplicate records. In this case, an error message is raised if you try to load a duplicate record.

This topic describes how to request duplicate checking for individual person records. It also describes how this option works in conjunction with the enterprise setting for duplicate checking.

Duplicate Checking for Individual Person Records

To request duplicate checking for a person record, you include the **PersonDuplicateCheck** attribute of the Worker object. This attribute can have one of the values shown in this table. The Description column identifies the attributes that the application uses in each case to identify duplicate records.

Value	Description
ORA_NONE or a blank value	No duplicate checking occurs
ORA_LN_FI_DOB_GEN_NID	Either the last name, first-name initial, date of birth, and gender or the national ID
ORA_LN_FI_DOB_NID	Either the last name, first-name initial, and date of birth or the national ID
ORA_LN_FN_DOB_GEN_NID	Either the last name, first name, date of birth, and gender or the national ID
ORA_NID_ONLY	The national ID

For example, if you set **PersonDuplicateCheck** to **ORA_LN_FI_DOB_NID**, then a duplicate record is identified if one of these situations occurs:

- The last name, first-name initial, and date of birth all match those of an existing person record.
- The national ID matches that of an existing person record.

Note: National ID values must be formatted. For example, to load the US social security number **987-65-4322**, you must include the hyphens. Don't specify the number as **987654322**. If you omit the formatting, which is country-specific, then duplicate person records aren't found when the checking is based on national identifier.

Enterprise Duplicate Checking

The enterprise option **Person Creation Service Duplicate Check** controls whether checks for duplicate person records occur by default when you load person records in bulk. If you exclude the **PersonDuplicateCheck** attribute of the Worker object, then the current setting of **Person Creation Service Duplicate Check** applies. If you include the **PersonDuplicateCheck** attribute, then the current setting of **Person Creation Service Duplicate Check** is ignored for the relevant person record.

Person Numbers in Worker Objects

Each person record, regardless of person type or number of work relationships, has a unique person number. Person numbers can be either generated automatically or entered manually, depending on the setting of the enterprise **Person Number Generation Method** option.

This topic describes ways of providing person numbers for uploaded Worker objects.

Automatically Generated Person Numbers

To generate person numbers automatically, you set the **Person Number Generation Method** option to either **Automatic Prior to Submission** or **Automatic Upon Final Save**. When you load person records to environments where person numbers are generated automatically, you supply no person number. A number is generated automatically on upload.

When you load a person record without a person number, you must supply a source key to identify the person record uniquely. You must use the same key for all child components of the Worker object using the **PersonId(SourceSystemId)** attribute and hint.

Manually Entered Person Numbers

When the **Person Number Generation Method** option is set to **Manual**, you must supply a person number in the Worker object.

Loading Legacy Numbers

You can load legacy numbers to environments where person numbers are generated automatically. Set the **Initial Person Number** enterprise option to the highest legacy person number plus one so that the legacy sequence continues.

Correcting Person Numbers

You can correct a person number, provided that **Person Number Generation Method** is set to **Manual**.

To correct a person number when the enterprise method is automatic generation, you can:

1. Note which of the two automatic methods, **Automatic Prior to Submission** and **Automatic Upon Final Save**, is being used.
2. Change the number-generation method temporarily to **Manual**.

   Note: You're recommended not to hire workers while the correction is being made.
3. Correct the person number.
4. Reinstate the original automatic number-generation method. Consider also whether the value of the **Initial Person Number** option must be updated to avoid conflict with the correction you made.
5. Run the **Update Person Search Keywords** process to ensure that person searches with person number as a keyword value are successful.

When you correct a person number, the correction applies to every date-effective update for the lifetime of the person record. This rule ensures that a person isn't identified using different person numbers at different times.

To identify the person record to correct, supply the source key, Oracle Fusion GUID, or Oracle Fusion surrogate ID. You can't supply the user key alone, as the person number is the user key.

Guidelines for Loading External Identifiers

The External Identifier component of the Worker object holds an identifier used by a third-party application, such as a time device or payroll application. This topic describes how to set some attributes of the External Identifier component.

External Identifiers at Person or Assignment Level

The external identifier can be captured:

- At the person level, by supplying only the person number
- At the assignment level, by supplying both the person number and the assignment number

External Identifier Types

The external identifier type is defined in lookup type ORA_PER_EXT_IDENTIFIER_TYPES. You can add lookup codes to this lookup type.

Date Values

The **DateFrom** and **DateTo** attributes include a time stamp. The time stamp ensures that you can assign an external identifier of a single type to a worker more than once on any day. For example, a person may be assigned a time-device badge identifier that he or she loses the same day. If you replace the badge identifier on the same day, then the time stamps differentiate the external identifiers.

External Identifier Sequence

The **ExternalIdentifierSequence** attribute is a required component of the user key and must be unique. You're recommended to set this attribute to **1** for the first record for a worker. Increment it by 1 for subsequent records.

External Identifier Example

This example Worker.dat file loads an external identifier for a person.

```
SET PURGE_FUTURE_CHANGES N
METADATA|ExternalIdentifier|ExternalIdentifierNumber|PersonNumber|ExternalIdentifierType|
ExternalIdentifierSequence
MERGE|ExternalIdentifier|rtyui45678|TestPer0TALTEST_9|Third-Party Payroll ID|1
```

Guidelines for Loading Person Names

The Person Name component of the Worker object holds both common and legislation-specific components of a person's name. This topic describes some decisions you must make when loading person names using HCM Data Loader.

Local and Global Names

Person names are always created with both a global and a local version. The name is created based on the name type that you supply, and the other name is derived automatically. For example, if you set the **NameType** attribute to **GLOBAL**, then the local name is derived. Alternatively, if you set **NameType** to **FR**, for example, then the global name is derived. Typically, only the global name is required and the application copies it to the local name automatically. However, if the person's name must be held in two different character sets, then supply the local name.

A core name field may not be available for a legislation. For example, Last Name Kanji isn't available for the Japanese legislation. In this case, the data is stored in one of the **NameInformation1** through **NameInformation14** attributes. **NameInformation1** through **NameInformation14** are reserved for this purpose. The person name styles table stores the mapping of the name style to the name attributes. The name style is the same for both the global and local-name records.

Name Attributes for Specific Legislations

To supply a name attribute for a specific legislation, use one of the **NameInformation15** through **NameInformation30** attributes. For example, to define the name attribute **Doing Business As** for the United States, you could use the **NameInformation15** attribute. You don't have to define flexfields for this requirement.

Supplying Date-Effective History

If you're loading date-effective history for a person name, then you must use source keys to identify the component. Otherwise, you can use the **PersonNumber**, **EffectiveStartDate**, and **NameType** user-key components.

Related Topics

- Oracle Fusion HCM: Person Name Validation (2146270.1)

Guidelines for Loading Person Address

The person address is a component of the Worker object. The person address is validated in these 2 ways:

- The first validation method is when the user creates an address in the responsive pages. The address is validated based on the geography data configuration as specified in the Manage Geographies task.

  However, this validation doesn't apply when address data is loaded using REST and HCM Data Loader. For these scenarios, you can run the Validate Geographies of Addresses Against Master Geographies ESS process with these parameters: Location Table Name = PER_ADDRESSES_F, Run Type = ALL, Country Code. This ESS process will give the list of addresses that don't match the validation. You can then correct these addresses.
- The second validation method is through Oracle-delivered business logic units. For example, for Netherlands, the business logic unit states that the postal code must follow the NNNNAA format (N is a number, A is either an uppercase or a lowercase letter). You can specify whether these validations should trigger when loading address data. You can do so by selecting the **Address Validation** check box for the specific country or territory in the Manage Features by Country or Territory task in the Setup and Maintenance work area. If **Address Validation** isn't selected, then the business logic unit validations won't be applied when address data is loaded using HCM Data Loader or through the UI.

For country-specific validations on the address attributes, it's recommended to check the localization documentation on Oracle Support.

Guidelines for Loading Person Images

The person image appears on various Oracle HCM Cloud application pages to identify a person's record. This topic describes aspects of loading the Person Image component of the Worker object using HCM Data Loader.

Image Type

You can set the **ImageType** attribute only to **PROFILE**. If you omit the **ImageType** attribute, then the type is **PROFILE** by default.

Image Size

The recommended size of the image is 90 pixels by 120 pixels. Images of different sizes are supported, but you're recommended to maintain an aspect ratio of 3 by 4 to reduce any distortion. No recommendation exists for the image resolution. The maximum size of the image file is 2 GB. However, you're recommended to store images at the smallest possible size, taking into account the suggested dimensions. The smaller the image, the better the performance of the image display in the application. Typically, the file size of an image of the recommended dimensions would be only 2 or 3 MB, even at high resolution.

Start Dates for Person Legislative Data

This topic explains how to specify start dates for person legislative data that you load using HCM Data Loader.

Person legislative data must start on the person's earliest effective start date. A person may have multiple work relationships in different legislations. In this case, all person legislative data must start on the person's earliest effective start date, not the start date of the work relationship. When a person starts a work relationship that's in a new legislation, the application:

- Creates a person legislative data record as of the person's earliest effective start date with no data.
- Performs an update of the person legislative data record as of the date of the creation of the new work relationship with the user-provided data. This approach ensures that the data is chronologically correct.

For example, consider the work relationships of the worker in this table.

Date	Action
2000	Hired by Vision Corporation in the UK
2005	Left Vision Corporation in the UK
2006	Joined US military
2009	Left US military and became a US veteran
2010	Hired by Vision Corporation in the US

In this example, in 2010 a US person legislative record is created as of 2000, the person's earliest effective start date. The US legislative record is updated as of 2010, the start date of the person's US work relationship. The update includes the descriptive flexfield details for US veteran status, as provided by the user.

Examples of Loading Work Relationship Changes

The Worker object provides several indicator attributes that simplify maintenance of work relationships. Use these indicators to request actions such as termination or a change of primary assignment. This topic provides examples of work-relationship changes that you request using these indicators.

> **Note:** These indicators aren't supported when you're supplying full date-effective history. They're provided solely for independent actions. Also, you can't provide multiple indicators on the same work relationship record. For example, you can't both terminate a work relationship (**TerminateWorkRelationshipFlag**) and perform a global transfer (**GlobalTransferFlag**) on the same work relationship at the same time.

Changing the Hire Date

The hire date is the start date of a work relationship. To change the hire date, provide the new date on the **NewStartDate** attribute of the Work Relationship component. You can load just the work relationship record. The employment terms and assignment records are maintained automatically.

This example changes the hire date for an existing worker.

```
SET PURGE_FUTURE_CHANGES N
METADATA|WorkRelationship|SourceSystemOwner|SourceSystemId|PersonId(SourceSystemId)|LegalEmployerName|
NewStartDate
MERGE|WorkRelationship|VISION|1009_POS|1009|Cox-6-CWB|2002/02/10
```

Terminating a Work Relationship

To terminate a work relationship, load the Work Relationship component with the **TerminateWorkRelationshipFlag** attribute set to **Y**. Specify the termination date on the **ActualTerminationDate** attribute. All associated assignments are automatically made inactive after the termination date.

This example terminates a work relationship.

```
SET PURGE_FUTURE_CHANGES N
METADATA|WorkRelationship|SourceSystemOwner|SourceSystemId|PersonId(SourceSystemId)|LegalEmployerName|
TerminateWorkRelationshipFlag|ActualTerminationDate|ActionCode|ReasonCode
MERGE|WorkRelationship|VISION|1009_POS|1009|Cox-6-CWB|Y|2015/10/02|RESIGNATION|RESIGN_PERSONAL
```

You can update the assignment status while terminating the work relationship by loading the work relationship and assignment rows in the Worker.dat file.

This example updates the assignment status while terminating the work relationship.

```
METADATA|WorkRelationship|SourceSystemOwner|SourceSystemId|PersonId(SourceSystemId)|LegalEmployerName|
ActualTerminationDate|ActionCode|ReasonCode
MERGE|WorkRelationship|VISION|1009_POS|1009|Cox-6-CWB|2015/10/02|RESIGNATION|RESIGN_PERSONAL
METADATA|WorkTerms|AssignmentNumber|PersonNumber|LegalEmployerName|DateStart|WorkerType|
ActionCode|ReasonCode|EffectiveStartDate|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|
AssignmentStatusTypeCode|BusinessUnitShortCode
MERGE|WorkTerms|ET955160008178426|955160008178426|Vision Corporation|2009/01/01|E|RESIGNATION|
RESIGN_PERSONAL|2015/10/03|4712/12/31|Y|1|INACTIVE_PROCESS|Vision ADB
METADATA|Assignment|AssignmentNumber|PersonNumber|LegalEmployerName|DateStart|WorkerType|ActionCode|
ReasonCode|WorkTermsNumber|EffectiveStartDate|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|
AssignmentStatusTypeCode|BusinessUnitShortCode|FLEX:PER_ASG_DF|hdlG(PER_ASG_DF=Global Data Elements)
```

```
MERGE|Assignment|E955160008178426|955160008178426|Vision Corporation|2009/01/01|E|RESIGNATION|
RESIGN_PERSONAL|ET955160008178426|2015/10/03|4712/12/31|Y|1|INACTIVE_PROCESS|Vision ADB|Global Data
 Elements|TestValue
```

Note: When a work relationship is terminated, the worker supervisor relationship is end-dated.

Correcting the Termination Date of a Work Relationship

To correct a work relationship termination date, supply a new **ActualTerminationDate** and set the **CorrectTerminationFlag** attribute to **Y**.

This example corrects the termination date for a terminated work relationship.

```
SET PURGE_FUTURE_CHANGES N
METADATA|WorkRelationship|SourceSystemOwner|SourceSystemId|PersonId(SourceSystemId)|LegalEmployerName|
CorrectTerminationFlag|ActualTerminationDate
MERGE|WorkRelationship|VISION|1009_POS|1009|Cox-6-CWB|Y|2015/10/08
```

Reversing a Termination

To reverse a termination, load the Work Relationship component with the **ReverseTerminationFlag** attribute set to **Y**.

This example reverses the termination of a work relationship.

```
SET PURGE_FUTURE_CHANGES N
METADATA|WorkRelationship|SourceSystemOwner|SourceSystemId|PersonId(SourceSystemId)|LegalEmployerName|
ReverseTerminationFlag
MERGE|WorkRelationship|VISION|1009_POS|1009|Cox-6-CWB|Y
```

Canceling a Work Relationship

To cancel a work relationship, use a DELETE instruction for the work relationship and set the **CancelWorkRelationshipFlag** attribute to **Y**.

This example deletes the specified work relationship and all its child records.

```
SET PURGE_FUTURE_CHANGES N
METADATA|WorkRelationship|SourceSystemOwner|SourceSystemId|PersonId(SourceSystemId)|
CancelWorkRelationshipFlag
DELETE|WorkRelationship|VISION|1008_POS|1008|Y
```

Changing the Primary Assignment or Work Relationship

When you load a primary work relationship, the existing primary work relationship automatically becomes nonprimary on the date that you supply on the **DateForPrimaryFlagChange** attribute.

To make a nonprimary work relationship primary, you load the work relationship to be made primary with the **PrimaryFlag** attribute set to **Y**. Specify the date when the work relationship becomes primary on the **DateForPrimaryFlagChange** attribute. The corresponding changes to employment terms and assignments occur automatically. The primary indicator for the existing primary work relationship is set to **N** automatically.

Employment Terms Override at Assignment

This topic describes how to set attribute values on employment terms and their associated assignments when you load them using HCM Data Loader.

The enterprise option **Allow Employment Terms Override at Assignment** controls whether assignment attributes inherited from employment terms can be overridden on the assignment. However, when you load employment terms, the associated assignments don't inherit attribute values from the employment terms. This table summarizes how to set attribute values on employment terms to avoid conflicting with the current setting of **Allow Employment Terms Override at Assignment**.

Allow Employment Terms Override at Assignment	Attribute Values on Employment Terms
No	Must match values on associated assignments
Yes	Can differ from values on associated assignments

Example of Loading Assignment Changes

You supply the assignment component of the Worker object when making assignment changes under the existing work relationship. This topic shows how to load assignment changes using HCM Data Loader.

Loading Assignment Changes

You can use HCM Data Loader to make the assignment changes for an existing worker. Keep these rules in mind:

- You must load the relevant Employment Terms and Assignment components in the same Worker.dat file.
- The values of the EffectiveStartDate and EffectiveEndDate attributes must be the same on the two components.
- The EffectiveStartDate value is the start date of the assignment change.
- The EffectiveEndDate value must be the end of time if the assignment record is the latest record.

The following example shows how to load the Employment Terms and Assignment components for an assignment change.

```
METADATA|WorkTerms|AssignmentNumber|PersonNumber|LegalEmployerName|DateStart|WorkerType|
ActionCode|ReasonCode|EffectiveStartDate|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|
AssignmentStatusTypeCode|BusinessUnitShortCode
MERGE|WorkTerms|ET955160008178426|955160008178426|Vision Corporation|2009/01/01|E|ASG_CHANGE||2020/01/01|
4712/12/31|Y|1|ACTIVE_PROCESS|Vision ADB
METADATA|Assignment|AssignmentNumber|PersonNumber|LegalEmployerName|DateStart|WorkerType|ActionCode|
ReasonCode|WorkTermsNumber|EffectiveStartDate|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|
AssignmentStatusTypeCode|BusinessUnitShortCode|FLEX:PER_ASG_DF|hdlG(PER_ASG_DF=Global Data Elements)
```

```
MERGE|Assignment|E955160008178426|955160008178426|Vision Corporation|2009/01/01|E|ASG_CHANGE||
ET955160008178426|2020/01/01|4712/12/31|Y|1|ACTIVE_PROCESS|Vision ADB|Global Data Elements|TestValue
```

Terminating an Assignment

To terminate an assignment, you must load the Assignment component with the TerminateAssignmentFlag attribute set to Y and specify the termination date for the TerminationDate attribute.

This example terminates an assignment.

```
METADATA|Assignment|ActionCode|AssignmentStatusTypeCode|AssignmentType|BusinessUnitShortCode|
EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|
WorkTermsAssignmentId(SourceSystemId)|SourceSystemOwner|SourceSystemId|TerminationDate|NotificationDate|
LastWorkingDate|ReviewUserAccess|TerminateAssignmentFlag
MERGE|Assignment|END_ASG|INACTIVE_PROCESS|E|Vision Corporation Enterprise|4712/12/31|Y|1|2022/07/03|
TEST_WT_ut_hdl18_02|HRC_SQLLOADER|TEST_ASG_ut_hdl18_02|2022/07/02||2022/07/02|A|Y
```

Note: If you don't provide the WorkTerm attribute along with the Assignment component on the same date, the following error message will be displayed: WorkTerms records required. For date effective records having TerminateAssignmentFlag as Y, the error won't be displayed. However, for date effective records without TerminateAssignmentFlag, the error will still apply.

Correcting an Assignment Termination

To correct an assignment termination, you must load the Assignment component with the CorrectAssignmentTerminationFlag attribute to Y.

This example corrects an assignment termination.

```
METADATA|Assignment|ActionCode|AssignmentStatusTypeCode|AssignmentType|BusinessUnitShortCode|
EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|
WorkTermsAssignmentId(SourceSystemId)|SourceSystemOwner|SourceSystemId|TerminationDate|NotificationDate|
LastWorkingDate|ReviewUserAccess|CorrectAssignmentTerminationFlag
MERGE|Assignment|END_ASG|INACTIVE_PROCESS|E|Vision Corporation Enterprise|4712/12/31|Y|1|2022/07/03|
TEST_WT_ut_hdl18_02|HRC_SQLLOADER|TEST_ASG_ut_hdl18_02|2022/07/02||2022/07/02|A|Y
```

Note:

- The LastWorkingDate attribute must be on or before the termination date. If the last working date exists and you don't provide a value, the existing last working date will be retained in case of postpone and auto calculated to the termination date in case of prepone. If you provide the last working date value, the new value will take precedence.
- The notification date will be retained in case of postpone if you don't provide a value. If you provide a value, the new value will take precedence. In case of prepone, you must provide a notification value. No error message will be displayed even if the notification date is after the termination date in case of postpone or prepone.
- If you correct the termination date for a terminated assignment to a specific date, the end date must be blank or end of time (12-31-4712).
- If you correct the existing assignment termination record data without modifying the termination date, you must pass the EffectiveStartDate and EffectiveEndDate attributes of the existing assignment termination record. In this case, your changes will be applicable for the first inactive record (termination record).
- Prepone will be allowed if there are future updates after the existing Termination or End Assignment inactive record. Future records will be deleted.
- Postpone won't be allowed if there are future updates after the existing Termination or End Assignment inactive record.

Reversing an Assignment Termination

To reverse a termination, you must load the Assignment component with the ReverseAssignmentTerminationFlag attribute set to Y.

This example reverses an assignment termination.

```
METADATA|Assignment|WorkTermsAssignmentId(SourceSystemId)|SourceSystemId|SourceSystemOwner|
ActionCode|ReasonCode|EffectiveStartDate|EffectiveEndDate|EffectiveSequence|EffectiveLatestChange|
ReverseAssignmentTerminationFlag
MERGE|Assignment|300100553121597|300100553121600|FUSION|ORA_EMPL_REV_TERMINATION||2022/05/02|4712/12/31|1|Y|
Y
```

Ending an Assignment

To end date an assignment, you must load an inactive assignment update with an ActionCode of type EMPL_END_ASG starting from the end date + 1 day.

This example end dates an assignment.

```
METADATA|Assignment|ActionCode|AssignmentStatusTypeCode|AssignmentType|BusinessUnitShortCode|
EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|
WorkTermsAssignmentId(SourceSystemId)|SourceSystemOwner|SourceSystemId
MERGE|Assignment|END_ASG|INACTIVE_PROCESS|E|Vision Corporation Enterprise|4712/12/31|Y|2|2022/07/03|
SSI_0004H_WTERM_1201_2|HRC_SQLLOADER|SSI_0004H_WASGN_1201_2
```

Note:

- It's recommended to use an ActionCode of type EMPL_END_TEMP_ASG to end date a temporary assignment.
- For temporary assignment loads, you can additionally use an active update on the same date for the suspended assignments. Additionally, you can load the active update in a separate DAT file.
- You can use UserKeys (AssignmentNumber, WorkTermsNumber) or surrogateIds (AssignmentId, WorkTermAssignmentId) in the DAT file instead of SourceSystemIds (SSIDs) mentioned in the example.

How You Update Line Manager for an Employee

You can update the line manager for an existing employee using HCM Data Loader. When loading this information, you must load the manager data before loading the employees reporting to them. You must include these discriminators in the Worker.dat file:

- WorkTerms
- Assignment
- AssignmentSupervisor

Both of these rows in the per_all_assignments_m table will need to be modified.

If you use user keys to uniquely identify the supervisor, you need these 3 attributes to retain original values.

- For current manager: ManagerAssignmentNumber
- For new manager: NewManagerAssignmentNumber and NewManagerPersonNumber

This example Worker.dat file updates the line manager for an existing worker.

```
METADATA|WorkTerms|EffectiveStartDate|EffectiveLatestChange|EffectiveSequence|EffectiveEndDate|ActionCode|
LegalEmployerName|DateStart|WorkerType|PersonNumber|AssignmentNumber
MERGE|WorkTerms|2021/05/01|Y|1|4712/12/31|MANAGER_CHANGE|Vision Corporation Limited|2017/03/19|E|235552|
E235552
METADATA|Assignment|EffectiveStartDate|EffectiveSequence|EffectiveLatestChange|EffectiveEndDate|ActionCode|
AssignmentNumber|WorkTermsNumber
MERGE|Assignment|2021/05/01|1|Y|4712/12/31|MANAGER_CHANGE|E235552|ET235552
METADATA|AssignmentSupervisor|EffectiveStartDate|EffectiveEndDate|AssignmentNumber|PrimaryFlag|
ManagerAssignmentNumber|ManagerType|NewManagerType|NewManagerPersonNumber|NewManagerAssignmentNumber
MERGE|AssignmentSupervisor|2021/05/01|4712/12/31|E235552|Y|E18344|LINE_MANAGER|LINE_MANAGER|16634|E16634
```

Note:

- You can have only one line manager at a point in time. You must send date effective changes for the new line manager.
- It's recommended to split the Worker.dat files and load the direct reports and managers separately, if the file size is huge. This is because HDL loads the data in chunks and it's possible the manager doesn't get created prior to the direct reports being loaded for them.

How You Perform a Global Transfer Using HCM Data Loader

This topic explains how to perform global transfers for both primary and nonprimary work relationships using HCM Data Loader. A global transfer is a permanent transfer from one legal employer to another.

It results in the termination of an existing work relationship and the creation of a new one. The termination date of the existing work relationship is one day prior to the start date of the new work relationship.

Performing a Global Transfer on a Primary Work Relationship

By default, the global transfer is applied to the primary work relationship. To perform a global transfer, you simply load the new Work Relationship component with all its child components. The new work relationship must have:

- A **GlobalTransferFlag** attribute set to **Y**
- An action code from the EMPL_GLB_TRANSFER action type
- A start date value that's the date of the global transfer

Don't include the work relationship that you're terminating. That work relationship is terminated automatically.

The termination and creation of the work relationships are validated to ensure data consistency. In particular, a worker can't be left with only an active, nonprimary work relationship. Therefore, if the worker has two active work relationships, you can't perform the global transfer on the primary one. You must make the primary work relationship nonprimary before you attempt the global transfer.

Performing a Global Transfer on a Nonprimary Work Relationship

A worker can have multiple nonprimary work relationships. Therefore, when you perform a global transfer of a nonprimary work relationship, you must supply additional information so that the correct work relationship is terminated. To perform the global transfer on a nonprimary work relationship, you must also supply:

- The parent Worker component.
- The nonprimary work relationship that you're terminating. However, don't include:
    - Any child records for the terminated work relationship
    - The **GlobalTransferFlag** attribute

These records are in addition to those that you supply for a global transfer of a primary work relationship.

Performing a Global Temporary Assignment

These are the restrictions that apply to the loading of global temporary assignments using HCM Data Loader.

When you create a global temporary assignment in the employment pages, two data rows are created:

- One row is created for the global temporary assignment.
- One row is added to the base assignment, which is suspended.

Both rows have the same action occurrence ID, which is defined by the application.

When you end the global temporary assignment in the UI, two further data rows are created:

- One row is created for the end global temporary assignment.
- One row is added to the base assignment, which becomes active on the day following the end date of the temporary assignment.

Both rows have the same action occurrence ID.

In HDL, the same UI behavior can't be replicated when you load data for a global temporary assignment. You need to provide data for both, the base and the new assigment in the base work relationship. Similarly, when you end a global temporary assignment, you need to provide data for the source and target work relationship.

> **Note:** If you end a global temporary assignment from the employment pages on an assignment row which was created using HDL, the source work relationship won't be automatically activated as the two work relationships don't share the same action occurence ID.

Updating Person-Name Legislation Codes

When you perform a global transfer, the new work relationship may have a legislation code that's new for the person. In this case, you may want to update the legislation code for the person name. The person-name legislation code is validated against all available work-relationship legislation codes. Therefore, the change to the work relationship must occur before the change to the person name. You can update the legislation code in the person name in one of these ways:

- Include the Worker object with no change in the same .dat file as the changes to the Person Name and Work Relationship components. In this case, changes to the work relationship are processed before changes to the person name.
- Process updates to the work relationship before those to the person name.

Linking Source And Target Assignments When Performing A Global Transfer

You can change a worker's legal employer with HCM Data Loader using the GlobalTransferFlag on the work relationship object. These new attributes are added to the assignment object:

- SourceAssignmentNumber
- SourceAssignmentId

If the user doesn't include the value for SourceAssignmentNumber or SourceAssignmentId, then the primary assignment ID from the work relationship that's being terminated is defaulted as the source assignment ID. These are the conditions for validating the source assignment:

- The assignment must belong to the person whose legal employer is being changed.
- The source assignment must start before the global transfer date.
- The source assignment must be active.

The example code below changes the legal employer for an existing worker who has only one work relationship and one assignment:

```
METADATA|WorkRelationship|SourceSystemOwner|SourceSystemId|PersonId(SourceSystemId)|LegalEmployerName|
PersonNumber|DateStart|WorkerType|PrimaryFlag|GlobalTransferFlag|ActionCode

MERGE|WorkRelationship|HRC_SQLLOADER|WR-HDLWorker_111_1|HDLWorker_111_1|Vision Corporation|HDLWorker_111_1|
1985/07/16|E|Y|Y|GLB_TRANSFER

METADATA|WorkTerms|SourceSystemOwner|SourceSystemId|ActionCode|ReasonCode|LegalEmployerName|
AssignmentNumber|AssignmentName|EffectiveStartDate|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|
PeriodOfServiceId(SourceSystemId)|PersonTypeCode|AssignmentStatusTypeCode|AssignmentType|BusinessUnitId|
NoticePeriod|PrimaryWorkTermsFlag|FreezeStartDate|FreezeUntilDate

MERGE|WorkTerms|HRC_SQLLOADER|WT-HDLWorker_111_1|GLB_TRANSFER||Vision Corporation|WT-NU-HDLWorker_111_1|
WT-NA-HDLWorker_111_1|1985/07/16|4712/12/31|N|1|WR-HDLWorker_111_1|Employee|ACTIVE_NO_PROCESS|ET|202|0|Y|
4712/12/31|1950/01/01

METADATA|Assignment|SourceSystemOwner|SourceSystemId|ActionCode|ReasonCode|LegalEmployerName|
AssignmentNumber|AssignmentName|EffectiveStartDate|EffectiveEndDate|EffectiveLatestChange|
EffectiveSequence|PeriodOfServiceId(SourceSystemId)|WorkTermsAssignmentId(SourceSystemId)|PersonTypeCode|
AssignmentStatusTypeCode|AssignmentType|BusinessUnitId|NoticePeriod|PrimaryAssignmentFlag|FreezeStartDate|
FreezeUntilDate|SourceAssignmentNumber

MERGE|Assignment|HRC_SQLLOADER|WA-HDLWorker_111_1|GLB_TRANSFER||Vision Corporation|WA-NU-HDLWorker_111_1|
WA-NA-HDLWorker_111_1|1985/07/16|4712/12/31|N|1|WR-HDLWorker_111_1|WT-HDLWorker_111_1|Employee|
ACTIVE_NO_PROCESS|E|202|0|Y|4712/12/31|1950/01/01|E34345588
```

Guidelines for Loading Additional Assignments

Whether you can load additional assignments for a worker depends on the employment model. This topic identifies, for each employment model, whether you can perform the Add Assignment action using HCM Data Loader.

Employment Model	Add Assignment
2 Tier - Single Assignment	No
2 Tier - Multiple Assignment	Yes
2 Tier - Single Contract - Single Assignment	No
2 Tier - Multiple Contract - Single Assignment	Yes

Employment Model	Add Assignment
3 Tier - Single Employment Terms - Single Assignment	No
3 Tier - Single Employment Terms - Multiple Assignment	Yes
3 Tier - Multiple Employment Terms - Single Assignment	No
3 Tier - Multiple Employment Terms - Multiple Assignment	Yes

The following example shows how to load an additional assignment. Both the Employment Terms and Assignment components are identified by user keys.

```
METADATA|WorkTerms|ActionCode|AssignmentNumber|AssignmentStatusTypeCode|AssignmentType|
BusinessUnitShortCode|DateStart|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|
LegalEmployerName|PersonNumber|PrimaryWorkTermsFlag|WorkerType
MERGE|WorkTerms|ADD_ASSIGN|ETTESTASSG03_562|ACTIVE_NO_PROCESS|ET|ZHRX-AE-Starter-BU|2016/01/01|4712/12/31|Y|
2|2017/08/01|CRFL RRF LE 2T|TESTASSG03_56|Y|E
METADATA|Assignment|ActionCode|AssignmentNumber|AssignmentStatusTypeCode|AssignmentType|
BusinessUnitShortCode|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|Frequency|
LegalEmployerName|PersonNumber|PrimaryAssignmentFlag|WorkTermsNumber|WorkerType
MERGE|Assignment|ADD_ASSIGN|ETESTASSG03_562|ACTIVE_NO_PROCESS|E|ZHRX-AE-Starter-BU|4712/12/31|Y|2|
2017/08/01|W|CRFL RRF LE 2T|TESTASSG03_56|Y|ETTESTASSG03_562|E
```

Guidelines for Loading Temporary Assignments

This topic describes restrictions that apply to the loading of temporary assignments using HCM Data Loader.

When you create a temporary assignment in the UI, two data rows are created:

1. One row is created for the temporary assignment.
2. One row is added to the base assignment, which is suspended.

Both rows have the same action occurrence ID, which is defined by the application.

When you end the temporary assignment, two further data rows are created:

1. One row is created to end the temporary assignment.
2. One row is added to the base assignment, which becomes active on the day following the end date of the temporary assignment.

Both rows have the same action occurrence ID.

However, when the temporary assignment is loaded using HDL, the two rows don't share the same action occurence ID. If you end a temporary assignment from the employment pages on an assignment row which was created using HDL, the source assignment won't be automatically activated as the two assignments don't share the same action occurence ID.

The following example shows how to load a temporary assignment. Both the Employment Terms and Assignment components are identified by user keys.

```
METADATA|WorkTerms|ActionCode|AssignmentNumber|AssignmentStatusTypeCode|AssignmentType|
BusinessUnitShortCode|DateStart|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|
LegalEmployerName|PersonNumber|PrimaryWorkTermsFlag|WorkerType
MERGE|WorkTerms|TEMP_ASG|ETTESTASSG03_59|SUSPEND_PROCESS|ET|ZHRX-AE-Starter-BU|2016/01/01|4712/12/31|Y|1|
2017/08/01|CRFL RRF LE 2T|TESTASSG03_59|Y|E
MERGE|WorkTerms|TEMP_ASG|ETTESTASSG03_59-2|ACTIVE_PROCESS|ET|ZHRX-AE-Starter-BU|2016/01/01|4712/12/31|Y|1|
2017/08/01|CRFL RRF LE 2T|TESTASSG03_59|N|E
METADATA|Assignment|ActionCode|AssignmentNumber|AssignmentStatusTypeCode|AssignmentType|
BusinessUnitShortCode|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|Frequency|
LegalEmployerName|PersonNumber|PrimaryAssignmentFlag|WorkTermsNumber|WorkerType
MERGE|Assignment|TEMP_ASG|ETESTASSG03_59|SUSPEND_PROCESS|E|ZHRX-AE-Starter-BU|4712/12/31|Y|1|2017/08/01|W|
CRFL RRF LE 2T|TESTASSG03_59|Y|ETTESTASSG03_59|E
MERGE|Assignment|TEMP_ASG|ETESTASSG03_59-2|ACTIVE_PROCESS|E|ZHRX-AE-Starter-BU|4712/12/31|Y|1|2017/08/01|W|
CRFL RRF LE 2T|TESTASSG03_59|N||ETTESTASSG03_59-2|E
```

Example of Adding an Assignment with a Shared Employment Contract

You supply the Contract component of the Worker object when adding additional worker assignments under the existing work relationship. This topic shows how to share an existing employment contract when adding assignments using HCM Data Loader.

Adding an Assignment with a Shared Employment Contract

You can use HCM Data Loader to share an employment contract across worker assignments. Keep these rules in mind:

- You must load the relevant Employment Terms, Assignment, and Contract components in the same Worker.dat file.
- The values of the **EffectiveStartDate** and **EffectiveEndDate** attributes must be the same on all three components.
- The **EffectiveStartDate** value is the start date of the shared employment contract. This value must be the projected end date of the last contract period, plus one day, meaning no gaps are permitted between contract periods.
- The **EffectiveEndDate** value must be the end of time if this contract record is the latest record. It must not be the contract end date.

The following example shows how to load the Employment Terms, Assignment, and Contract components for a shared employment contract. You can use this HCM Data Loader file format for associating an existing contract with the newly created worker assignment:

```
METADATA|Worker|PersonNumber|DateOfBirth|StartDate|EffectiveStartDate|EffectiveEndDate|ActionCode|
SourceSystemId|SourceSystemOwner
MERGE|Worker|202001001_DB08|1953/11/01|2020/01/01|2020/01/01|4712/12/31|ADD_ASSIGN|PERSON_202001001_DB08|
VISION
METADATA|WorkRelationship|PersonNumber|DateStart|LegalEmployerName|WorkerType|PrimaryFlag|SourceSystemOwner
MERGE|WorkRelationship|202001001_DB08|2020/01/01|VISION_GHR_2TSCSA|E|Y|WR_202001001_DB08|VISION
METADATA|WorkTerms|EffectiveStartDate|EffectiveSequence|EffectiveLatestChange|ActionCode|
AssignmentStatusTypeCode|AssignmentType|WorkerType|LegalEmployerName|BusinessUnitShortCode|AssignmentNumber|
PeriodOfServiceId(SourceSystemId)|PersonNumber|SourceSystemOwner|SourceSystemId|PrimaryWorkTermsFlag
```

```
MERGE|WorkTerms|2021/01/01|1|Y|ADD_ASSIGN|ACTIVE_PROCESS|ET|E|VISION_GHR_2TSCSA|Vision City Operations|
WT_202001002_DB08|WR_202001001_DB08|202001001_DB08|VISION|WT_202001002_DB08|N
METADATA|Assignment|EffectiveStartDate|EffectiveSequence|EffectiveLatestChange|ActionCode|
AssignmentStatusTypeCode|AssignmentType|WorkerType|LegalEmployerName|PrimaryAssignmentFlag|PrimaryFlag|
BusinessUnitShortCode|WorkTermsAssignmentId(SourceSystemId)|SourceSystemOwner|SourceSystemId|ContractNumber
MERGE|Assignment|2021/01/01|1|Y|ADD_ASSIGN|ACTIVE_PROCESS|E|E|VISION_GHR_2TSCSA|N|N|Vision City Operations|
WT_202001002_DB08|VISION|ASG_202001002_DB08|ContractNUM
```

Note:

- The contract number that's provided in the HCM Data Loader file must be valid for the work relationship for which the new assignment is being created.
- If you want to add a contract to an existing assignment that doesn't have a contract, ensure that you pass the assignment record and the corresponding contract record in the same DAT file.

Example of Adding a Contract

You can supply the Contract component of the Worker object for existing workers.

This topic shows how to load a contract for an existing worker using HCM Data Loader.

Loading a Contract

You can use HCM Data Loader to give an existing worker a contract extension. Keep these rules in mind:

- You must load the relevant Employment Terms, Assignment, and Contract components in the same Worker.dat file.
- The values of the EffectiveStartDate and EffectiveEndDate attributes must be the same on all three components.
- The EffectiveStartDate value is the start date of the extension period. This value must be the projected end date of the last contract period, plus one day, meaning no gaps are permitted between contract periods.
- The EffectiveEndDate value must be the end of time if this contract record is the latest record. It must not be the contract end date.
- You can't delete a contract record using HCM Data Loader.
- You can create or edit a contract only with an assignment.

The following example shows how to load the Employment Terms, Assignment, and Contract components for a contract.

```
METADATA|WorkTerms|EffectiveStartDate|EffectiveEndDate|EffectiveSequence|EffectiveLatestChange|ActionCode|
AssignmentStatusTypeCode|AssignmentType|WorkerType|LegalEmployerName|BusinessUnitShortCode|AssignmentNumber|
PeriodOfServiceId(SourceSystemId)|PersonId(SourceSystemId)|PersonNumber|SourceSystemOwner|SourceSystemId|
ProjectedEndDate
MERGE|WorkTerms|2001/11/01|4712/12/31|1|Y|HIRE|ACTIVE_PROCESS|ET|E|GHR_Vision_LE_11|Vision City
 Operations|ET202001001_CON11|WR_202001001_CON11|PERSON_202001001_CON11|202001001_CON11|HRC_SQLLOADER|
WT_202001001_CON11|

METADATA|Assignment|EffectiveStartDate|EffectiveEndDate|EffectiveSequence|EffectiveLatestChange|ActionCode|
AssignmentStatusTypeCode|AssignmentType|WorkerType|LegalEmployerName|PrimaryAssignmentFlag|PrimaryFlag|
BusinessUnitShortCode|PersonId(SourceSystemId)|WorkTermsAssignmentId(SourceSystemId)|SourceSystemOwner|
SourceSystemId|ProjectedEndDate
```

```
MERGE|Assignment|2001/11/01|4712/12/31|1|Y|HIRE|ACTIVE_PROCESS|E|E|GHR_Vision_LE_11|Y|Y|Vision City
 Operations|PERSON_202001001_CON11|WT_202001001_CON11|HRC_SQLLOADER|ASG_202001001_CON11|

METADATA|Contract|EffectiveStartDate|EffectiveEndDate|Duration|DurationUnits|ExtensionPeriod|
ExtensionPeriodUnits|AssignmentNumber|ContractType|PersonNumber|SourceSystemOwner|SourceSystemId
MERGE|Contract|2001/11/01|4712/12/31|10|M|||ET202001001_CON11||202001001_CON11|HRC_SQLLOADER|
CONT_ET202001001_CON11
```

Example of Loading a Contract Extension

You supply the Contract component of the Worker object when creating workers, if appropriate.

However, you can load a contract extension for a worker who has a contract. This topic shows how to load a contract extension using HCM Data Loader.

Loading a Contract Extension

You can use HCM Data Loader to give an existing worker a contract extension. Keep these rules in mind:

- You must load the relevant Employment Terms, Assignment, and Contract components in the same Worker.dat file.
- The values of the **EffectiveStartDate** and **EffectiveEndDate** attributes must be the same on all three components.
- The **EffectiveStartDate** value is the start date of the extension period. This value must be the projected end date of the last contract period, plus one day, meaning no gaps are permitted between contract periods.
- The **EffectiveEndDate** value must be the end of time if this contract record is the latest record. It must not be the contract end date.
- You can specify the extension period by updating the **ExtensionPeriod** and **ExtensionPeriodUnits** attributes of the Contract component.

  In this case, the new projected end date is calculated automatically.
- The value of the **ActionCode** attribute on the Employment Terms and Assignment components must be associated with the action type EMPL_CONTRACT_EXTN.
- You can't delete a contract extension record using HCM Data Loader.

The following example shows how to load the Employment Terms, Assignment, and Contract components for a contract extension.

```
METADATA|WorkTerms|SourceSystemOwner|PersonId(SourceSystemId)|PeriodOfServiceId(SourceSystemId)|
SourceSystemId|AssignmentId|EffectiveStartDate|EffectiveEndDate|EffectiveSequence|EffectiveLatestChange|
ActionCode
MERGE|WorkTerms|FUSION|300100162474267|300100162474475|300100162474482|300100162474482|2011/01/31|
4712/12/31|1|Y|CONTRACT_EXTENSION
METADATA|Assignment|SourceSystemOwner|PersonId(SourceSystemId)|PeriodOfServiceId(SourceSystemId)|
WorkTermsAssignmentId(SourceSystemId)|SourceSystemId|AssignmentId|EffectiveStartDate|EffectiveEndDate|
EffectiveSequence|EffectiveLatestChange|ActionCode
MERGE|Assignment|FUSION|300100162474267|300100162474475|300100162474482|300100162474487|300100162474487|
2011/01/31|4712/12/31|1|Y|CONTRACT_EXTENSION
METADATA|Contract|SourceSystemOwner|PersonId(SourceSystemId)|PersonNumber|AssignmentId(SourceSystemId)|
SourceSystemId|ContractId|EffectiveStartDate|EffectiveEndDate|ExtensionPeriod|ExtensionPeriodUnits
MERGE|Contract|FUSION|300100162474267|960000000000615|300100162474482|300100162474490|300100162474490|
2011/01/31|4712/12/31|100|D
```

Tip: If you want to correct an existing contract row (for example, contract extension), ensure that you pass the assignment record and the corresponding contract record in the same DAT file. You can only correct the contract end date and contract duration attributes for the latest contract record (second extension record in the table). This is because any changes that you make to these attributes in the initial records aren't propagated to the subsequent future records.

Automatic Calculation of FTE Values for Workers

The full-time equivalent (FTE) value is the result of dividing assignment working hours by standard working hours, which are typically those of a full-time worker.

For example, if the assignment working hours are 10 and the standard working hours are 40, then the FTE value is 0.25. When you edit assignment working hours in the user interface, FTE is calculated automatically. When you load assignment records using HCM Data Loader, you can specify FTE values on the Assignment Work Measure component. Alternatively, you can request that FTE values be calculated automatically. This topic describes how to request automatic calculation of FTE values.

Calculating FTE Values Automatically

To request automatic calculation of FTE values for all assignments in a Worker.dat file, include the following SET instruction in the file:

```
SET CALCULATE_FTE Y
```

When you include this instruction, the following rules apply:

- You must not include an Assignment Work Measure component for the assignment in the Worker.dat file.
- No more than one assignment work measure record can exist for the assignment in Oracle HCM Cloud.
- Both assignment working hours and standard working hours must exist for the assignment throughout the relevant time period. Otherwise, no calculation of FTE values occurs for the assignment.

You may find automatic calculation of FTE values helpful if:

- You're loading many assignment records that include assignment working hours.
- You aren't loading FTE values in the Assignment Work Measure component.

By including the `SET CALCULATE_FTE Y` instruction in the .dat file, you can generate FTE values for all of the assignments.

Tip: You can perform a data load with the sole purpose of calculating FTE values. Simply include the user keys of the assignments in a Worker.dat file that includes the `SET CALCULATE_FTE Y` instruction.

The `SET CALCULATE_FTE Y` instruction generates FTE values only. It doesn't generate headcount values.

Related Topics

- File Line Instructions and File Discriminators
- The SET Instruction

Create a Default Working-Hour Pattern

A working-hour pattern defines the working hours, start time, and end time for each day of the week. When you load assignment records using HCM Data Loader, you can request that a default working-hour pattern record be created automatically.

Creating a Default Working-Hour Pattern

To create a default working-hour pattern for all assignments in a Worker.dat file, include this instruction in the Worker.dat file:

```
SET CREATE_DEFAULT_WORKING_HOUR_PATTERN Y
```

You may find automatic creation of default working-hour patterns helpful if you're loading many assignment records but aren't loading the Working Hour Pattern component of the Worker object. Once this default record exists, you can update it with information for each worker by loading the Working Hour Pattern component of the Worker object.

Related Topics

- File Line Instructions and File Discriminators
- The SET Instruction

Examples of Loading Worker Working Hour Pattern

You can define shift types and working hours for your workers by loading working hour patterns with HCM Data Loader.

If the daily working hours are a fixed number of hours per day, use Elapsed as the Work Shift Type to load. If you need to track the start and end times per day, use Time as the Work Shift Type to load. Let's look at some Worker.dat files that create worker hour patterns.

Load Working Hour Pattern by Shift Type

To get started with working hour patterns, see the topic Create a Default Working-Hour Pattern. Then, you can use the following examples to load worker hours with the work shift type of your choice.

Let's say you want to define the specific times of day in a worker's daily shift. To do this, load the Working Hour Pattern using Time as the Work Shift Type.

```
METADATA|WorkTerms|ActionCode|AssignmentStatusTypeCode|AssignmentType|EffectiveEndDate|
EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|AssignmentNumber|LegalEmployerName|
PrimaryWorkTermsFlag|DateStart|PersonNumber|WorkerType
MERGE|WorkTerms|HIRE|ACTIVE_PROCESS|ET|4712/12/31|Y|1|2019/03/29|E51323|V1_LE0227|Y|2019/03/13|51323|E
METADATA|Assignment|ActionCode|AssignmentStatusTypeCode|AssignmentType|EffectiveEndDate|
EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|WorkTermsNumber|AssignmentNumber|PrimaryFlag|
PrimaryAssignmentFlag|HourlySalariedCode|PersonNumber
MERGE|Assignment|HIRE|ACTIVE_PROCESS|E|4712/12/31|Y|1|2019/03/29|ET966169008889921|E51323|Y|Y|Salaried|51323
METADATA|WorkingHourPattern|Object|AssignmentNumber|EffectiveStartDate|WorkDayDefinitionCode|
WorkingHoursType|WorkShiftType|SatEndTime|SatStartTime|FriStartTime|FriEndTime|MonStartTime|MonEndTime|
SunEndTime|SunStartTime|ThuEndTime|ThuStartTime|TueEndTime|TueStartTime|WedStartTime|WedEndTime
```

```
MERGE|WorkingHourPattern|ASSIGNMENT|E51323|2019/03/29|TIME ENTRY STOP DAY|ORA_WORK_WEEK|ORA_TIME|18:00|
10:00|10:00|18:00|10:00|18:00|18:00|10:00|18:00|10:00|18:00|10:00|10:00|18:00
```

If you would rather record how many hours worked per shift, you can set Work Shift Type as Elapsed.

```
METADATA|WorkTerms|ActionCode|AssignmentStatusTypeCode|AssignmentType|EffectiveEndDate|
EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|AssignmentNumber|LegalEmployerName|
PrimaryWorkTermsFlag|DateStart|PersonNumber|WorkerType
MERGE|WorkTerms|HIRE|ACTIVE_PROCESS|ET|4712/12/31|Y|1|2019/03/29|E51323|V1_LE0227|Y|2019/03/13|51323|E
METADATA|Assignment|ActionCode|AssignmentStatusTypeCode|AssignmentType|EffectiveEndDate|
EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|WorkTermsNumber|AssignmentNumber|PrimaryFlag|
PrimaryAssignmentFlag|HourlySalariedCode|PersonNumber
MERGE|Assignment|HIRE|ACTIVE_PROCESS|E|4712/12/31|Y|1|2019/03/29|ET966169008889921|E51323|Y|Y|Salaried|51323
METADATA|WorkingHourPattern|Object|AssignmentNumber|EffectiveStartDate|WorkDayDefinitionCode|
WorkingHoursType|WorkShiftType|WedHours|TueHours|ThuHours|SunHours|MonHours|FriHours|SatHours
MERGE|WorkingHourPattern|ASSIGNMENT|E51323|2019/03/29|TIME ENTRY STOP DAY|ORA_WORK_WEEK|ORA_ELAPSED|8|8|8|8|
8|8|8
```

Example of Loading PeopleGroup for a Worker

If you want to assign workers to groups, like certain pension plans, you can use the People Group key flexfield to do that. Here's an example on how to load the People Group flexfield.

Load a People Group Flexfield

Let's say you want to assign people to pension plans. First, you add the 'PeopleGroup' attribute to a METADATA Assignment record in your Worker.dat file. Then add a MERGE Assignment record with the value for the people group that you want. In this example, 'PeopleGroup' and its value appear at the end of the metadata and merge entries, respectively.

```
METADATA|Worker|PersonId|EffectiveStartDate|EffectiveEndDate|PersonNumber|BloodType|CorrespondenceLanguage|
StartDate|DateOfBirth|DateOfDeath|CountryOfBirth|RegionOfBirth|TownOfBirth|ApplicantNumber|
WaiveDataProtectFlag|CategoryCode|SourceSystemOwner|SourceSystemId|GUID|ActionCode|ReasonCode|
SourceRefTableName=PER_ALL_PEOPLE_F|SourceRef001=PERSON_ID|SourceRef002=PERSON_NUMBER|
SourceRefName003=START_DATE
MERGE|Worker||2000/01/01|4712/12/31|Worker_DL_101|||2000/01/01|1970/01/01|||||Y||HRC_SQLLOADER|
SSIW_DL_101_1234_1||HIRE||||Worker_DL_101|2000/01/01
METADATA|PersonName|PersonNameId|EffectiveStartDate|EffectiveEndDate|PersonId|PersonNumber|
LegislationCode|NameType|FirstName|MiddleNames|LastName|Honors|KnownAs|PreNameAdjunct|MilitaryRank|
PreviousLastName|Suffix|Title|CharSetContext|SourceSystemOwner|SourceSystemId|GUID|NameInformation1|
NameInformation2|NameInformation3|NameInformation4|NameInformation5|NameInformation6|NameInformation7|
NameInformation8|NameInformation9|NameInformation10|NameInformation11|NameInformation12|NameInformation13|
NameInformation14|NameInformation15|NameInformation16|NameInformation17|NameInformation18|
NameInformation19|NameInformation20|NameInformation21|NameInformation22|NameInformation23|NameInformation24|
NameInformation25|NameInformation26|NameInformation27|NameInformation28|NameInformation29|
NameInformation30|SourceRefTableName=PER_PERSON_NAMES_F|SourceRef001=PERSON_ID|SourceRef002=PERSON_NUMBER|
SourceRef003=NAME_TYPE|SourceRef004=LAST_NAME
MERGE|PersonName||2000/01/01|4712/12/31||Worker_DL_101|US|GLOBAL|Worker_DL_101 FName|MN|Worker_DL_101 LName|
Doctor|Worker_DL_101||12|Prev LName|Suffix|Mr.|US|HRC_SQLLOADER|SSIW_DL_101_NAME_1234_1||Worker_DL_101|NI2|
NI3|NI4|NI5|NI6|NI7|NI8|NI9|NI10|NI11|NI12|NI13|NI14|NI15|NI16|NI17|NI18|NI19|NI20|NI21|NI22|NI23|NI24|NI25|
NI26|NI27|NI28|NI29|NI30|||Worker_DL_101|GLOBAL|Worker_DL_101 LName
METADATA|PersonLegislativeData|PersonLegislativeId|EffectiveStartDate|EffectiveEndDate|
PersonId|PersonNumber|LegislationCode|HighestEducationLevel|MaritalStatus|MaritalStatusDate|Sex|
SourceSystemOwner|SourceSystemId|GUID|SourceRefTableName=PER_PEOPLE_LEGISLATIVE_F|SourceRef001=PERSON_ID|
SourceRef002=LEGISLATION_CODE
MERGE|PersonLegislativeData||2000/01/01|4712/12/31||Worker_DL_101|US||Married|2002/01/01|Male|HRC_SQLLOADER|
SSIW_DL_101_LEG_1234_1||||US
METADATA|WorkRelationship|LegalEmployerSeniorityDate|ActualTerminationDate|LegalEntityId|Comments|
EnterpriseSeniorityDate|LastWorkingDate|DateStart|NotifiedTerminationDate|OnMilitaryServiceFlag|
```

```
PeriodOfServiceId|PersonId|PrimaryFlag|ProjectedTerminationDate|RehireAuthorizerPersonId|RehireAuthorizor|
RehireReason|RevokeUserAccess|WorkerNumber|PersonNumber|LegalEmployerName|RehireRecommendationFlag|
WorkerType|GUID|SourceSystemId|SourceSystemOwner|NewStartDate|SourceRefTableName=PER_PERIODS_OF_SERVICE|
SourceRef001=LEGAL_ENTITY_ID|SourceRef002=LEGISLATION_CODE|SourceRef003=DATE_START
MERGE|WorkRelationship|||||||2000/01/01|||||Y||||||55857|Worker_DL_101|Vision Corporation||E||
SSIW_DL_101_WREL_1234_2|HRC_SQLLOADER|||US|2000/01/01
METADATA|WorkTerms|ActionCode|AssignmentId|AssignmentName|AssignmentNumber|AssignmentStatusTypeCode|
AssignmentStatusTypeId|AssignmentType|BargainingUnitCode|BillingTitle|BusinessUnitId|BusinessUnitShortCode|
ContractId|DateProbationEnd|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|
WorkerCategory|AssignmentCategory|EstablishmentId|ReportingEstablishment|ExpenseCheckSendToAddress|
GradeId|GradeCode|GradeLadderPgmId|GradeLadderPgmName|HourlySalariedCode|InternalBuilding|InternalFloor|
InternalLocation|InternalMailstop|InternalOfficeNumber|JobId|JobCode|LocationId|LocationCode|
ManagerFlag|NormalHours|Frequency|NoticePeriod|NoticePeriodUOM|OrganizationId|DepartmentName|
PeriodOfServiceId(SourceSystemId)|PersonId|PersonNumber|DateStart|WorkerType|LegalEmployerName|
PersonTypeId|PersonTypeCode|PositionCode|PositionOverrideFlag|PrimaryWorkTermsFlag|ProbationPeriod|
ProbationUnit|ProjectedEndDate|ProjectedStartDate|ProposedWorkerType|ReasonCode|RetirementAge|
RetirementDate|SpecialCeilingStepId|SpecialCeilingStep|StepEntryDate|SystemPersonType|TaxAddressId|
EndTime|StartTime|VendorSiteId|WorkAtHomeFlag|WorkTermsAssignmentId|VendorId|FreezeStartDate|
FreezeUntilDate|GUID|SourceSystemOwner|SourceSystemId|CategoryCode|TaxReportingUnitId|TaxReportingUnit|
SourceRefTableName=PER_ALL_ASSIGNMENTS_M|SourceRef001=ASSIGNMENT_NAME|SourceRef002=ASSIGNMENT_NUMBER|
SourceRef003=ASSIGNMENT_TYPE|DefaultExpenseAccount
MERGE|WorkTerms|HIRE||TestWT_Worker_DL_101|TestWTNum_Worker_DL_101|ACTIVE_PROCESS||ET|||202||||
4712/12/31|Y|1|2000/01/01|WC|FR|||||||||||||||||||||||||SSIW_DL_101_WREL_1234_2|||||Vision Corporation||
Employee|||Y|||||||||||||||||||||||HRC_SQLLOADER|SSIW_DL_101_WTERM_1234_1|||||TestWT_Worker_DL_101|
TestWTNum_Worker_DL_101|ET|01-300-1150-00
METADATA|Assignment|ActionCode|AssignmentId|EffectiveStartDate|EffectiveEndDate|EffectiveSequence|
EffectiveLatestChange|AssignmentType|AssignmentName|AssignmentNumber|AssignmentStatusTypeCode|
AssignmentStatusTypeId|BargainingUnitCode|BillingTitle|BusinessUnitId|BusinessUnitShortCode|
DateProbationEnd|WorkerCategory|AssignmentCategory|EstablishmentId|ReportingEstablishment|
ExpenseCheckSendToAddress|GradeId|GradeCode|GradeLadderPgmId|GradeLadderPgmName|HourlySalariedCode|
InternalBuilding|InternalFloor|InternalLocation|InternalMailstop|InternalOfficeNumber|JobId|JobCode|
LabourUnionMemberFlag|LocationCode|LocationId|ManagerFlag|NormalHours|Frequency|NoticePeriod|
NoticePeriodUOM|OrganizationId|PeriodOfServiceId(SourceSystemId)|PersonNumber|PersonId|DateStart|
WorkerType|LegalEmployerName|PersonTypeCode|PersonTypeId|PositionCode|PositionId|PositionOverrideFlag|
PrimaryAssignmentFlag|ProbationPeriod|ProbationUnit|ProjectTitle|ProjectedEndDate|ProjectedStartDate|
ProposedWorkerType|ReasonCode|RetirementAge|RetirementDate|SpecialCeilingStep|SpecialCeilingStepId|
SystemPersonType|TaxAddressId|EndTime|StartTime|VendorSiteId|WorkAtHomeFlag|WorkTermsNumber|
WorkTermsAssignmentId(SourceSystemId)|DepartmentName|VendorId|FreezeStartDate|FreezeUntilDate|
GUID|SourceSystemId|SourceSystemOwner|CategoryCode|SourceRefTableName=PER_ALL_ASSIGNMENTS_M|
SourceRef001=ASSIGNMENT_NAME|SourceRef002=ASSIGNMENT_NUMBER|SourceRef003=ASSIGNMENT_TYPE|
DefaultExpenseAccount|PeopleGroup
MERGE|Assignment|HIRE||2000/01/01|4712/12/31|1|Y|E|TestWANM_Worker_DL_101|TestWANum_Worker_DL_101|
ACTIVE_PROCESS||||202|||WC|FR|||||||||||||||||||||||SSIW_DL_101_WREL_1234_2|||2000/01/01||Vision
 Corporation|Employee|||||Y|||||||||||||||||TestWTNum_Worker_DL_101|SSIW_DL_101_WTERM_1234_1||||||
SSIW_DL_101_WASGN_1234_1|HRC_SQLLOADER|||TestWANM_Worker_DL_101|TestWANum_Worker_DL_101|E|01-300-1150-00|A
```

Delete a People Group Flexfield

Let's say, for GDPR or other compliance purposes, you must remove values from a 'PeopleGroup' flexfield segment and disable it. First, enable the segment if it's currently disabled. Next, change the values for the segment in your HCM Data Loader file to empty. Don't use spaces, and make sure the values are empty. Upload your HCM Data Loader file and confirm that the segments appear empty. Finally, disable the segment, if applicable.

Example of Loading a Default Expense Account for a Worker

For every worker who uses Oracle Expenses Cloud, a default expense account must be defined. The Default Expense Account key flexfield is available on the Assignment component of the Worker object.

This topic shows how to load the default expense account for a worker on the worker assignment.

Loading the Default Expense Account on the Assignment

If the expense account is configured with multiple segments, then provide the concatenated value. Separate the segment values using the separator configured for the expense account, as shown in this example.

```
METADATA|Assignment|AssignmentNumber|EffectiveStartDate|EffectiveEndDate|EffectiveSequence|
EffectiveLatestChange|ActionCode|BusinessUnitId|DepartmentName|LocationCode|PrimaryAssignmentFlag|JobCode|
GradeCode|DefaultExpenseAccount|WorkTermsAssignmentId(SourceSystemId)|SourceSystemOwner|SourceSystemId
MERGE|Assignment|E10020592|2012/08/13|2015/04/30|1|Y|MIGRATED|300000001934525|5005 - ICT|GB|N|ICT Manager|M|
5101.0000.3030.0000.231200.0000.00000000.0000.0000|10020592_WORKTERMS|VISION|10020592_ASSIGNMENT
MERGE|Assignment|E10020592|2015/05/01|4712/12/31|1|Y|MIGRATED|300000001934525|5005 - ICT|GB|Y|ICT Manager|M|
5101.0000.3030.0000.231200.0000.00000000.0000.0000|10020592_WORKTERMS|VISION|10020592_ASSIGNMENT
```

Example of Loading a Tax Reporting Unit for a Worker

The Tax Reporting Unit (TRU) ID and TRU fields are added to the worker data (worker.dat) file when creating workers, work relationships, or assignments using HCM Data Loader.

You can pass the TRU ID or TRU name in the worker.dat file when loading workers in these scenarios:

Hire

```
METADATA|Worker|EffectiveStartDate|EffectiveEndDate|PersonNumber|StartDate|DateOfBirth|WaiveDataProtectFlag|
SourceSystemOwner|SourceSystemId|ActionCode
MERGE|Worker|2000/01/01|4712/12/31|||2000/01/01|1970/01/01|Y|HRC_SQLLOADER|SSI_0004A_0123_1|HIRE
METADATA|PersonName|EffectiveStartDate|EffectiveEndDate|PersonId(SourceSystemId)|LegislationCode|NameType|
FirstName|MiddleNames|LastName|CharSetContext|SourceSystemOwner|SourceSystemId
MERGE|PersonName|2000/01/01|4712/12/31|SSI_0004A_0123_1|US|GLOBAL|Worker_new_SSI_0004A_0123_1FName||
Worker_new_SSI_0004A_0123_1LName|US|HRC_SQLLOADER|SSI_0004A_NAME_0123_1
METADATA|WorkRelationship|LegalEntityId|ActualTerminationDate|DateStart|PrimaryFlag|WorkerNumber|
PersonId(SourceSystemId)|WorkerType|SourceSystemId|SourceSystemOwner
MERGE|WorkRelationship|40010|||2000/01/01|Y|||SSI_0004A_0123_1|E|SSI_0004A_WREL_0123_1|HRC_SQLLOADER
METADATA|WorkTerms|ActionCode|AssignmentStatusTypeCode|AssignmentType|BusinessUnitId|EffectiveEndDate|
EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|SystemPersonType|WorkerCategory|
PeriodOfServiceId(SourceSystemId)|SourceSystemOwner|SourceSystemId
MERGE|WorkTerms|HIRE|ACTIVE_PROCESS|ET|202|4712/12/31|Y|1|2000/01/01|EMP|WC|SSI_0004A_WREL_0123_1|
HRC_SQLLOADER|SSI_0004A_WTERM_0123_1
METADATA|Assignment|ActionCode|TaxReportingUnitId|AssignmentId|EffectiveStartDate|EffectiveEndDate|
EffectiveSequence|EffectiveLatestChange|AssignmentType|SystemPersonType|BusinessUnitId|
PeriodOfServiceId(SourceSystemId)|DateStart|PrimaryAssignmentFlag|WorkTermsAssignmentId(SourceSystemId)|
SourceSystemId|SourceSystemOwner
MERGE|Assignment|HIRE|1017||2000/01/01|4712/12/31|1|Y|E|EMP|202|SSI_0004A_WREL_0123_1|2000/01/01|Y|
SSI_0004A_WTERM_0123_1|SSI_0004A_WASGN_0123_1|HRC_SQLLOADER
```

Create Work Relationship

```
METADATA|WorkRelationship|LegalEmployerSeniorityDate|LegalEntityId|EnterpriseSeniorityDate|DateStart|
OnMilitaryServiceFlag|PeriodOfServiceId|PersonId(SourceSystemId)|PrimaryFlag|PersonNumber|LegalEmployerName|
WorkerType|GUID|SourceSystemId|SourceSystemOwner|NewStartDate|ActionCode
MERGE|WorkRelationship||||2011/01/01|Y||SSI_0004M_0123_1|N||Vision Corporation|C||SSI_0004M_0123_2|
HRC_SQLLOADER||ADD_CWK_WORK_RELATION
METADATA|WorkTerms|ActionCode|AssignmentId|AssignmentName|AssignmentNumber|AssignmentStatusTypeCode|
AssignmentStatusTypeId|AssignmentType|BargainingUnitCode|BillingTitle|BusinessUnitId|BusinessUnitShortCode|
ContractId|DateProbationEnd|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|EffectiveStartDate|
WorkerCategory|AssignmentCategory|ReportingEstablishment|EstablishmentId|ExpenseCheckSendToAddress|
GradeId|GradeCode|GradeLadderPgmId|HourlySalariedCode|InternalBuilding|InternalFloor|InternalLocation|
InternalMailstop|InternalOfficeNumber|JobId|JobCode|LocationId|LocationCode|ManagerFlag|NormalHours|
Frequency|NoticePeriod|NoticePeriodUOM|OrganizationId|DepartmentName|PeriodOfServiceId(SourceSystemId)|
PersonId|PersonNumber|LegalEmployerName|DateStart|WorkerType|PositionCode|PositionOverrideFlag|
PrimaryWorkTermsFlag|ProbationPeriod|ProbationUnit|ProjectedEndDate|ProjectedStartDate|ProposedWorkerType|
ReasonCode|RetirementAge|RetirementDate|SpecialCeilingStepId|SpecialCeilingStep|StepEntryDate|
TaxAddressId|EndTime|StartTime|VendorSiteId|WorkAtHomeFlag|WorkTermsAssignmentId|VendorId|FreezeStartDate|
FreezeUntilDate|GUID|SourceSystemOwner|SourceSystemId|CategoryCode|PersonTypeCode
MERGE|WorkTerms|ADD_CWK_WORK_RELATION||||ACTIVE_PROCESS||CT|||202|Vision Corporation Enterprise|||
4712/12/31|Y|1|2011/01/01|WC|FR|Progress Master|||||||||||||||||||||||SSI_0004M_0123_2|||||||||||||||
2040/12/12|||||||N|||||HRC_SQLLOADER|SSI_0004M_WTERM_0123_2||Contingent Worker
METADATA|Assignment|ActionCode|AssignmentId|TaxReportingUnitId|AssignmentName|AssignmentNumber|
AssignmentStatusTypeCode|AssignmentStatusTypeId|AssignmentType|BargainingUnitCode|BillingTitle|
BusinessUnitId|BusinessUnitShortCode|DateProbationEnd|EffectiveEndDate|EffectiveLatestChange|
EffectiveSequence|EffectiveStartDate|WorkerCategory|AssignmentCategory|EstablishmentId|
ExpenseCheckSendToAddress|GradeId|GradeCode|GradeLadderPgmId|HourlySalariedCode|InternalBuilding|
InternalFloor|InternalLocation|InternalMailstop|InternalOfficeNumber|JobId|JobCode|LocationId|
LocationCode|ManagerFlag|NormalHours|Frequency|NoticePeriod|NoticePeriodUOM|OrganizationId|
DepartmentName|PeriodOfServiceId|PersonId|PersonNumber|LegalEmployerName|DateStart|WorkerType|
PositionCode|PositionOverrideFlag|PrimaryAssignmentFlag|ProbationPeriod|ProbationUnit|
ProjectedEndDate|ProjectedStartDate|ProposedWorkerType|ReasonCode|RetirementAge|RetirementDate|
SpecialCeilingStepId|SpecialCeilingStep|TaxAddressId|EndTime|StartTime|VendorSiteId|WorkAtHomeFlag|
WorkTermsAssignmentId(SourceSystemId)|VendorId|FreezeStartDate|FreezeUntilDate|GUID|SourceSystemOwner|
SourceSystemId|CategoryCode
MERGE|Assignment|ADD_CWK_WORK_RELATION||300100201616314|||ACTIVE_PROCESS||C|||202|Vision Corporation
 Enterprise||4712/12/31|Y|1|2011/01/01|BC|FR|||||||||||||||||||||||||||||||Y|||||||||||||||
SSI_0004M_WTERM_0123_2|||||HRC_SQLLOADER|SSI_0004M_WASGN_0123_2|
```

Add Assignment

```
METADATA|Assignment|ActionCode|TaxReportingUnitId|AssignmentId|AssignmentStatusTypeCode|AssignmentType|
BusinessUnitId|BusinessUnitShortCode|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|
EffectiveStartDate|WorkerCategory|AssignmentCategory|JobCode|ManagerFlag|NormalHours|Frequency|
DepartmentName|PrimaryAssignmentFlag|WorkAtHomeFlag|WorkTermsAssignmentId(SourceSystemId)|SourceSystemOwner|
SourceSystemId|CategoryCode|PersonTypeCode
MERGE|Assignment|ADD_ASSIGN|1017||ACTIVE_PROCESS|E||Vision Corporation Enterprise|4712/12/31|Y|1|2010/01/01|
WC|FR|JOBCD2320|N|40|W|Engineering R+D|Y|N|22334455|HRC_SQLLOADER|33445566_1||Employee
```

Global Transfer

```
METADATA|WorkRelationship|SourceSystemOwner|SourceSystemId|PersonId(SourceSystemId)|LegalEmployerName|
PersonNumber|DateStart|WorkerType|PrimaryFlag|GlobalTransferFlag|ActionCode
MERGE|WorkRelationship|HRC_SQLLOADER|WR-HDLWorker_111_1|HDLWorker_111_1|Vision Corporation|HDLWorker_111_1|
1985/07/16|E|Y|Y|GLB_TRANSFER
METADATA|WorkTerms|SourceSystemOwner|SourceSystemId|ActionCode|ReasonCode|LegalEmployerName|
AssignmentNumber|AssignmentName|EffectiveStartDate|EffectiveEndDate|EffectiveLatestChange|EffectiveSequence|
PeriodOfServiceId(SourceSystemId)|PersonTypeCode|AssignmentStatusTypeCode|AssignmentType|BusinessUnitId|
NoticePeriod|PrimaryWorkTermsFlag|FreezeStartDate|FreezeUntilDate
```

```
MERGE|WorkTerms|HRC_SQLLOADER|WT-HDLWorker_111_1|GLB_TRANSFER||Vision Corporation|WT-NU-HDLWorker_111_1|
WT-NA-HDLWorker_111_1|1985/07/16|4712/12/31|N|1|WR-HDLWorker_111_1|Employee|ACTIVE_NO_PROCESS|ET|202|0|Y|
4712/12/31|1950/01/01
METADATA|Assignment|SourceSystemOwner|SourceSystemId|ActionCode|TaxReportingUnitId|ReasonCode|
LegalEmployerName|AssignmentNumber|AssignmentName|EffectiveStartDate|EffectiveEndDate|EffectiveLatestChange|
EffectiveSequence|PeriodOfServiceId(SourceSystemId)|WorkTermsAssignmentId(SourceSystemId)|PersonTypeCode|
AssignmentStatusTypeCode|AssignmentType|BusinessUnitId|NoticePeriod|PrimaryAssignmentFlag|FreezeStartDate|
FreezeUntilDate
MERGE|Assignment|HRC_SQLLOADER|WA-HDLWorker_111_1|GLB_TRANSFER|1017||Vision Corporation|WA-NU-
HDLWorker_111_1|WA-NA-HDLWorker_111_1|1985/07/16|4712/12/31|N|1|WR-HDLWorker_111_1|WT-HDLWorker_111_1|
Employee|ACTIVE_NO_PROCESS|E|202|0|Y|4712/12/31|1950/01/01
```

Note:

- In the worker.dat file, you must add the TaxReportingUnitId field only in the assignment row and not in the work terms row.
- This data will be used in future by Payroll. Therefore, it's recommended not to supply data for the TRU ID and TRU fields in the worker.dat file unless it's ready to be used in Payroll. When this feature is activated for Payroll, you will see the feature under the Workforce Rewards What's New under your country.

FAQs for Loading Workers

How do I load person phones?

You must include any formatting characters in the value that you specify on the **PhoneNumber** attribute of the Person Phone component.

For example, to enter the US phone 650.555.0185 in the UI, you enter **650** in the **Area Code** field and **555-0185** in the **Number** field. When loading this number using HCM Data Loader, you must specify the value **555-0185** on the **PhoneNumber** attribute.

Formatting requirements vary by legislation. Always include the formatting characters, such as hyphen or period, specified for your legislation when loading person phones using HCM Data Loader.

If you include the **LegislationCode** attribute, then any **CountryCodeNumber** value that you supply must be valid for the specified legislation code.

How do I load person national identifiers?

You must include any formatting characters in the value that you specify on the **NationalIdentifierNumber** attribute of the Person National Identifier component.

For example, to load the US social security number **987-65-4322**, you must include the hyphens. Don't specify the number as **987654322**. If you omit the formatting, which is country-specific, then duplicate person records aren't found when the checking is based on national identifier.

How do I specify probation values on an assignment?

You can supply either the **ProbationPeriod** and **ProbationUnit** values or the **DateProbationEnd** value.

- If you supply the **ProbationPeriod** and **ProbationUnit**, then the **DateProbationEnd** value is calculated automatically.
- If you supply the **DateProbationEnd**, then the **ProbationPeriod** and **ProbationUnit** values are calculated automatically.
- If you supply values for all three attributes, then an error is raised.

Until update 17D, if you supplied the **DateProbationEnd**, then the **ProbationPeriod** and **ProbationUnit** values weren't calculated automatically. In addition, if you supplied all three attributes, then the **DateProbationEnd** value was calculated automatically and the values you supplied were discarded. These errors are now fixed.

How do I load a pending worker without providing a projected start date?

You can't load a pending worker without providing a projected start date.

Can I add a date-effective update to an existing pending worker record using HCM Data Loader?

No, you can only make corrections to an existing pending worker record.

What value is used if no value is passed for the position override flag?

If no value is passed for the position override flag, the application defaults the value to N.

Can I load employment data or prevent loading when employment changes are pending approval?

Yes. You can do it using the **HR_DISABL_PENDING_APPROVALS_CHECK_IN_ HCM_DATA_LOADER** profile option. To enable loading, at the **Site** level, set **Profile Value** to **No**.

To prevent loading, set the **Profile Value** to **Yes**. If the profile option is set to **No**, and you load data with the pending approval transaction, then the approval transaction fails.

Can checklists be allocated automatically to workers created using HCM Data Loader?

Checklists aren't automatically allocated to workers created using HCM Data Loader. You need to either load the allocated checklist object for the worker using HCM Data Loader or use the Allocate Checklists page to manually allocate the checklist.

4 Loading Worker-Related Objects

Examples of Loading Assignment Eligible Jobs

For any worker assignment, you can identify additional jobs for which the worker is eligible. You can use this feature for workers who must report time for the additional jobs, for example.

You can also use it simply to track the additional jobs of workers who have multiple jobs. This topic provides examples showing how to create and update Assignment Eligible Job components of the Worker object using HCM Data Loader.

Before you load Assignment Eligible Job components:

- The assignment job must exist in the target environment.
- The job family must exist in the target environment if the relief type is **Derived**.

  Relief type can be either **Manual** or **Derived**.

  - When the relief type is **Derived**, you can add jobs belonging to the job family of the worker's assignment only. You can specify the **ReliefType** value using either the meaning **Derived** or the lookup code **ORA_D**.
  - When the relief type is **Manual**, you can add any job. You must specify **ManualRate** and **Frequency** values. You can specify the **ReliefType** value using either the meaning **Manual** or the lookup code **ORA_M**.

Also, the start date of the eligible job must not be before the assignment start date.

You load assignment eligible jobs data in the Worker.dat file for processing by HCM Data Loader. You can verify this data on the Manage Eligible Jobs page for a selected worker.

Creating Assignment Eligible Jobs

This example Worker.dat file creates an assignment eligible job for a worker using source keys. **ReliefType** is **Manual**.

```
METADATA|AssignmentEligibleJob|AssignmentId(SourceSystemId)|JobId(SourceSystemId)|ToDate|FromDate|
SourceSystemOwner|SourceSystemId|ReliefType|ManualRate|Frequency
MERGE|AssignmentEligibleJob|1031101972|100000011571171|2000/04/01|2000/01/01|VISION|TEST_RC_ut0001|ORA_M|
11.7|Hourly
```

This example Worker.dat file creates an assignment eligible job for a worker using source keys. **ReliefType** is **Derived**.

```
METADATA|AssignmentEligibleJob|AssignmentId(SourceSystemId)|JobId(SourceSystemId)|ToDate|FromDate|
SourceSystemOwner|SourceSystemId|ReliefType
MERGE|AssignmentEligibleJob|1031101860|100000011571171|2000/04/01|2000/01/01|VISION|TEST_RC_ut0002|ORA_D
```

This example Worker.dat file creates assignment eligible jobs for a worker using user keys. **ReliefType** is **Manual**. The user key attributes for this component are **AssignmentNumber**, **JobCode**, **BusinessUnitShortCode**, and **EffectiveStartDate**.

```
METADATA|AssignmentEligibleJob|AssignmentNumber|JobCode|ToDate|FromDate|BusinessUnitShortCode|ReliefType|
ManualRate|Frequency
MERGE|AssignmentEligibleJob|TEST_ASG_ut0001|JOBOPMANCORE|2000/04/01|2000/01/01|Vision Corporation
 Enterprise|Manual|11.7|Hourly
```

Updating Assignment Eligible Jobs

You can update only the value of the **ToDate** attribute of an assignment eligible job. This example Worker.dat file updates a worker's assignment eligible job using source keys.

```
METADATA|AssignmentEligibleJob|AssignmentId(SourceSystemId)|JobCode|ToDate|FromDate|BusinessUnitShortCode|
SourceSystemOwner|SourceSystemId|ReliefType|ManualRate|Frequency
MERGE|AssignmentEligibleJob|1031101972|JOBOPMANCORE|2001/04/01|2000/01/01|Vision Corporation Enterprise|
VISION|TEST_RC_ut0001|Manual|11.7|Hourly
```

This example Worker.dat file updates a worker's assignment eligible job using user keys.

```
METADATA|AssignmentEligibleJob|AssignmentNumber|JobCode|ToDate|FromDate|BusinessUnitShortCode|ReliefType|
ManualRate|Frequency
MERGE|AssignmentEligibleJob|TEST_ASG_ut0001|JOBOPMANCORE|2001/04/01|2000/01/01|Vision Corporation
 Enterprise|Manual|11.7|Hourly
```

Guidelines for Loading Seniority Dates

A seniority date is the date on which the calculation of a worker's length of service with the enterprise is based. A worker's status, rank, or entitlement to benefits may depend on his or her seniority.

The seniority date can be based on the worker's time in a particular entity, such as a legal employer or job. Therefore, you can configure seniority rules based on values including collective agreement, country, department, enterprise, grade, job, legal employer, location, position, and union. This topic describes some factors that affect how you load seniority dates using HCM Data Loader.

Versions of the Seniority Dates Functionality

Three versions of the seniority dates functionality, referred to as V1, V2, and V3, exist. The following table identifies:

- The user interface task that you use to manage each version
- Whether you can update the seniority dates for each version using HCM Data Loader

Version	User Interface Task	Update Using HCM Data Loader	Worker Component
V1	Manage Work Relationship **Note:** You can update only the enterprise and legal employer seniority dates.	Yes	Work Relationship
V2	Manage Seniority Dates, without fast formula support	No	Not applicable
V3	Manage Seniority Dates, with fast formula support	Yes	Seniority Date

Note: After you migrate to V3 seniority dates, you can't use either V1 or V2. Therefore, you can't load seniority dates on the Work Relationship component. This topic applies to V3 seniority dates.

For more information, see the document Seniority Dates - Comparison between Different Seniority Dates Versions (https://community.oracle.com/customerconnect/discussion/631037) on Customer Connect.

Running the Calculate Seniority Dates Process

You can update but not create V3 seniority records for workers using HCM Data Loader. You must run the **Calculate Seniority Dates** process to create default seniority records for workers, based on configured seniority date rules, before you update them. After updating seniority records using HCM Data Loader, you can verify them for a worker on the Manage Seniority Dates page.

Required Attributes

The following table identifies attribute values that you must include in the Seniority Date component of the Worker object to associate it with a seniority date rule. These requirements depend on the level at which the seniority date rule is configured.

Level of Seniority Date Rule	Required Attributes
Work relationship	One of: • PeriodOfServiceId • LegalEmployerName, DateStart, and WorkerType
Assignment	One of: • AssignmentId • AssignmentNumber
Person	One of: • PersonId • PersonNumber

Updating Seniority Dates Using Source Keys

As you can't create V3 seniority records using HCM Data Loader, they have default source keys where the source-system owner is FUSION. Therefore, to update seniority dates using source keys, you must:

- Extract their default source keys using the HCM extract Integration Object User Key Extract.
- Update them using the Source Key object so that all source-key references for a single Seniority Date component have the same source-system owner.

Related Topics

- Examples of Loading Seniority Date Adjustments

Examples of Loading Seniority Date Adjustments

A seniority date is the date on which the calculation of a worker's length of service with the enterprise is based. This topic provides examples showing how to update V3 seniority dates using HCM Data Loader.

The following table identifies attributes that you can use to identify the entity on which the seniority date is based. You can use alternative key types, where available. For example, to identify a job you can use the user key (**JobCode** and **BusinessUnitShortCode**) instead of the **JobId** value.

Entity	Identifying Attribute
Bargaining unit	BargainingUnitCode
Collective agreement	CollectiveAgreementId
Department	DepartmentId
Enterprise	BusinessUnitShortCode
Grade	GradeId
Grade step	GradeStepId
Job	JobId
Legal employer	LegalEntityId
Legislation	LegislationCode
Location	LocationId
Position	PositionId
Union membership	UnionId

Updating Seniority Dates Using User Keys

The following example Worker.dat files update a worker's seniority date based on various assignment attributes. They use user keys to reference the Seniority Date component. As these examples show, you must include the **ManualAdjustmentComments** attribute when you make a manual adjustment to the seniority date. In these examples, the seniority date rule is configured at the person level.

This example updates a worker's seniority date based on the assignment.

```
METADATA|SeniorityDate|SeniorityDateCode|PersonNumber|EntryDate|EffectiveStartDate|EffectiveEndDate|
BusinessUnitShortCode|ManualAdjustmentDays|ManualAdjustmentComments|JobCode|AssignmentNumber
MERGE|SeniorityDate|JUN_A|9990101|2012/01/01|2015/01/01|4712/12/31|Vision Corporation Enterprise|20|Data
 Correction|JOBCD7|E9990401
```

This example updates a worker's seniority date based on the department.

```
METADATA|SeniorityDate|SeniorityDateCode|PersonNumber|EntryDate|EffectiveStartDate|EffectiveEndDate|
BusinessUnitShortCode|ManualAdjustmentDays|ManualAdjustmentComments|DepartmentName
MERGE|SeniorityDate|JUN_DEPARTMENT|9990101|2012/01/01|2015/01/01|4712/12/31|Vision Corporation Enterprise|
20|Data Correction|Commercial Sales
```

This example updates a worker's seniority date based on the enterprise.

```
METADATA|SeniorityDate|SeniorityDateCode|PersonNumber|EntryDate|EffectiveStartDate|EffectiveEndDate|
BusinessUnitShortCode|ManualAdjustmentDays|ManualAdjustmentComments
MERGE|SeniorityDate|JUN_ET|9990101|2012/01/01|2015/01/01|4712/12/31|Vision Corporation Enterprise|20|Data
 Correction
```

This example updates a worker's seniority date based on the grade.

```
METADATA|SeniorityDate|SeniorityDateCode|PersonNumber|EntryDate|EffectiveStartDate|EffectiveEndDate|
BusinessUnitShortCode|ManualAdjustmentDays|ManualAdjustmentComments|GradeCode
MERGE|SeniorityDate|JUN_GRADE|9990101|2012/01/01|2015/01/01|4712/12/31|Vision Corporation Enterprise|20|Data
 Correction|DHRX-AE-IC1
```

This example updates a worker's seniority date based on the job.

```
METADATA|SeniorityDate|SeniorityDateCode|PersonNumber|EntryDate|EffectiveStartDate|EffectiveEndDate|
BusinessUnitShortCode|ManualAdjustmentDays|ManualAdjustmentComments|JobCode
MERGE|SeniorityDate|JUN_JOB|9990101|2012/01/01|2015/01/01|4712/12/31|Vision Corporation Enterprise|20|Data
 Correction|JOBCD7
```

This example updates a worker's seniority date based on the job but uses the seniority date rule name rather than its code.

```
METADATA|SeniorityDate|SeniorityDateCode|PersonNumber|EntryDate|EffectiveStartDate|EffectiveEndDate|
BusinessUnitShortCode|ManualAdjustmentDays|ManualAdjustmentComments|JobCode
MERGE|SeniorityDate|SeniorityDateJUnit Person Job|9990101|2012/01/01|2015/01/01|4712/12/31|Vision
 Corporation Enterprise|20|Data Correction|JOBCD7
```

This example updates a worker's seniority date based on the legal employer.

```
METADATA|SeniorityDate|SeniorityDateCode|PersonNumber|EntryDate|EffectiveStartDate|EffectiveEndDate|
BusinessUnitShortCode|ManualAdjustmentDays|ManualAdjustmentComments|LegalEmployerName
MERGE|SeniorityDate|JUN_LEMP|9990101|2012/01/01|2015/01/01|4712/12/31|Vision Corporation Enterprise|20|Data
 Correction|Vision Corporation
```

This example updates a worker's seniority date based on the location.

```
METADATA|SeniorityDate|SeniorityDateCode|PersonNumber|EntryDate|EffectiveStartDate|EffectiveEndDate|
BusinessUnitShortCode|ManualAdjustmentDays|ManualAdjustmentComments|LocationCode
MERGE|SeniorityDate|JUN_LOCATION|9990101|2012/01/01|2015/01/01|4712/12/31|Vision Corporation Enterprise|20|
Data Correction|V1-_NEW_YORK_CITY_0_2450399170046
```

This example updates a worker's seniority date based on the position.

```
METADATA|SeniorityDate|SeniorityDateCode|PersonNumber|EntryDate|EffectiveStartDate|EffectiveEndDate|
BusinessUnitShortCode|ManualAdjustmentDays|ManualAdjustmentComments|PositionCode
MERGE|SeniorityDate|JUN_POSITION|9990101|2012/01/01|2015/01/01|4712/12/31|Vision Corporation Enterprise|20|
Data Correction|POSCD66
```

This example updates a worker's seniority date based on the work relationship.

```
METADATA|SeniorityDate|SeniorityDateCode|PersonNumber|EntryDate|EffectiveStartDate|EffectiveEndDate|
BusinessUnitShortCode|ManualAdjustmentDays|ManualAdjustmentComments|JobCode|WorkerType|LegalEmployerName|
DateStart
MERGE|SeniorityDate|JUN_WR|9990101|2012/01/01|2015/01/01|4712/12/31|Vision Corporation Enterprise|20|Data
 Correction|JOBCD7|E|Vision Corporation|2001/05/05
```

Related Topics

- Guidelines for Loading Seniority Dates

Examples of Loading Seniority Hours

You calculate seniority in hours for hourly paid workers. So that the application can perform seniority calculations for these workers, you load their seniority hours using HCM Data Loader.

You can obtain the seniority hours from various sources, such as Oracle Time and Labor, Oracle Global Payroll, or a third-party application. Any worker assignment for which you load seniority hours must exist and be active as of the start date of the hours that you're loading. It must also be paid hourly for the entire period for which you're loading data.

> **Tip:** You can verify the loaded data for a worker on the Manage Seniority Dates page. To verify loaded data for multiple workers, you can query the PER_SENIORITY_HOURS table.

This topic provides examples showing how to create, update, and delete the Seniority Hour component of the Worker object using HCM Data Loader.

Creating Seniority Hours

This example Worker.dat file shows how to create the Seniority Hour component of the Worker object using source keys.

```
METADATA|SeniorityHour|AssignmentId(SourceSystemId)|ToDate|FromDate|Hours|SourceSystemOwner|SourceSystemId
MERGE|SeniorityHour|1031101972|2001/01/01|2000/01/01|500|VISION|UT00214
```

|

This example Worker.dat file shows how to create the Seniority Hour component of the Worker object using user keys. The user key attributes for this component are **AssignmentNumber** and **FromDate**.

```
METADATA|SeniorityHour|AssignmentNumber|ToDate|FromDate|Hours
MERGE|SeniorityHour|E00214|2001/01/01|2000/01/01|500
```

Updating Seniority Hours

This example Worker.dat file shows how to update the Seniority Hour component of the Worker object using source keys.

```
METADATA|SeniorityHour|AssignmentId(SourceSystemId)|ToDate|FromDate|Hours|SourceSystemOwner|SourceSystemId
MERGE|SeniorityHour|1031101972|2001/01/01|2000/01/01|600|VISION|UT00214
```

This example Worker.dat file shows how to update the Seniority Hour component of the Worker object using user keys.

```
METADATA|SeniorityHour|AssignmentNumber|ToDate|FromDate|Hours
MERGE|SeniorityHour|E00214|2001/01/01|2000/01/01|600
```

Deleting Seniority Hours

This example Worker.dat file shows how to delete the Seniority Hour component of a Worker object using source keys.

```
METADATA|SeniorityHour|AssignmentId(SourceSystemId)|SourceSystemOwner|SourceSystemId
DELETE|SeniorityHour|1031101972|VISION|UT00214
```

This example Worker.dat file shows how to delete the Seniority Hour component of a Worker object using user keys.

```
METADATA|SeniorityHour|AssignmentNumber|ToDate|FromDate|Hours
DELETE|SeniorityHour|E00214|2001/01/01|2000/01/01|500
```

Guidelines for Loading Allocated Checklists

An allocated checklist is a specific instance of a checklist template that's assigned to one or more workers. The checklist contains tasks that workers must perform.

Rules for Allocated Checklists

These rules apply when you load allocated checklists using HCM Data Loader:

- Both the checklist template and the worker to whom the checklist is allocated must exist in the target environment.
- Workflow notifications aren't triggered for the checklist tasks.
- The allocation happens immediately, regardless of any allocation date specified in the .dat file.
- If the AllowAutoAllocation attribute is set as Y (default value), then eligibility is evaluated at checklist and task level.
- When a task is added or any other updates are made to allocated checklists and tasks, then eligibility is not evaluated.

Related Topics

- Examples of Loading an Allocated Checklist

Examples of Loading an Allocated Checklist

This topic shows how to load an allocated checklist using HCM Data Loader.

Allocating Only Checklist Template

A checklist can be allocated by providing only the checklist template details and person details. You don't need to provide all task details, as the details are retrieved from the checklist template definition. This is the preferred option for bulk allocation.

Note: HDL is run using an elevated mode with FUSION_APPS_HCM_ESS_APP_ID as user. This isn't related to any person. InitiatorPersonNumber is an alternative that's similar to the option on the user interface and can be used to determine the initiator.

This example AllocateChecklist.dat file allocates a checklist template using user keys. InitiatorPersonNumber will need to be passed in case the performer or owner is the Initiator.

```
METADATA|AllocateChecklist|PersonNumber|ChecklistName|ChecklistInstance|ChecklistCategory|Country|
TemplateChecklistName|TemplateChecklistCategory MERGE|AllocateChecklist|8153762|Envision Offboard|1|Off
 Boarding||Envision Offboard|Off Boarding
```

This example AllocatedChecklist.dat file allocates the checklist and creates all child components as defined in the checklist template using source keys.

```
METADATA|AllocateChecklist|SourceSystemOwner|SourceSystemId|PersonNumber|ChecklistName|ChecklistInstance|
ChecklistCategory|Country|TemplateChecklistName|TemplateChecklistCategory|ChecklistId(SourceSystemId)
MERGE|AllocateChecklist|VISION|ALLOC-8153812|8153812|Onboarding Checklist|8153812|ONBOARD|United States|||
Onboarding Checklist01
```

Creating an Enterprise Onboarding Allocated Checklist

This example AllocateChecklist.dat file allocates an enterprise onboarding checklist using user keys. You create an Enterprise Onboarding checklist using a three-step process.

Note: Since this isn't auto-allocation there is no reference to the checklist template.

Step 1: Create Step Checklist and Tasks

```
COMMENT - PASS1 Create Step Checklists and corresponding tasks.
METADATA|AllocateChecklist|ChecklistName|ChecklistCategory|Country|ChecklistInstance|ChecklistStatus|
PersonNumber|AllowAutoAllocation|InitiatorPersonNumber
MERGE|AllocateChecklist|Vision Enterprise Onboarding Step|ORA_ONB_ENT_ONBOARDING_STEP|United States|1|INI|
7109026|N|8153756
METADATA|AllocateChecklistTask|TaskName|ChecklistName|ChecklistCategory|Country|ChecklistInstance|
PersonNumber|MandatoryFlag|Status|TaskPerformerPersonNumber|TaskOwnerPersonNumber|ActionType|PerformerRole|
OwnerRole|InitiatorPersonNumber
MERGE|AllocateChecklistTask|Apply for credit card|Vision Enterprise Onboarding Step|
ORA_ONB_ENT_ONBOARDING_STEP|United States|1|7109026|N|INI|8153757|8153757|ORA_CHK_MANUAL|ORA_ADHOC_USER|
ORA_CHK_ADHOC_USER|8153756
```

Step 2: Create Enterprise Onboarding Master Checklist and Link to Step Checklist

```
COMMENT - PASS2 Create Enterprise Onboarding and its related artifacts.
METADATA|AllocateChecklist|ChecklistName|Country|ChecklistCategory|ChecklistInstance|ChecklistStatus|
PersonId|PersonNumber|Description|AllowAutoAllocation|ActionDate|AllocationDate|EventDate|
InitiatorPersonNumber
MERGE|AllocateChecklist|Vision Enterprise Onboarding|United States|ORA_ONB_ENT_ONBOARDING|1|INI||7109026|
Test Enterprise Onboarding|N|2020/01/01 00:00:00|2020/01/01 00:00:00|2020/01/01 00:00:00|8153756
COMMENT - TASK in Enterprise Onboarding corresponding to Step Checklist Created in PASS1
METADATA|AllocateChecklistTask|TaskName|ChecklistName|Country|ChecklistCategory|ChecklistInstance|
PersonNumber|Description|MandatoryFlag|DetailAllocChecklistName|DetailAllocChecklistCategory|
DetailAllocChecklistCountry|DetailAllocChecklistInstance|AbsAllocatedChecklistName|
AbsAllocatedChecklistCategory|AbsAllocatedChecklistCountry|AbsAllocatedChecklistInstance|ActionType|
DetailChecklistId|Status
MERGE|AllocateChecklistTask|First 30 days|Vision Enterprise Onboarding|United States|ORA_ONB_ENT_ONBOARDING|
1|7109026|Desc Task parent|Y|Vision Enterprise Onboarding Step|ORA_ONB_ENT_ONBOARDING_STEP|United States|
1|||||ORA_CHK_CHECKLIST||INI
```

Step 3: Update Task Details in Enterprise Onboarding Step Checklist

```
METADATA|AllocateChecklist|ChecklistName|ChecklistCategory|Country|ChecklistInstance|ChecklistStatus|
PersonNumber|AllowAutoAllocation|InitiatorPersonNumber
MERGE|AllocateChecklist|Vision Enterprise Onboarding Step|ORA_ONB_ENT_ONBOARDING_STEP|United States|1|INI|
7109026|N|8153756
METADATA|AllocateChecklistTask|TaskName|ChecklistName|ChecklistCategory|Country|ChecklistInstance|
PersonNumber|MandatoryFlag|Status|TaskPerformerPersonNumber|TaskOwnerPersonNumber|ActionType|
PerformerRole|OwnerRole|InitiatorPersonNumber|AbsAllocatedChecklistName|AbsAllocatedChecklistCategory|
AbsAllocatedChecklistCountry|AbsAllocatedChecklistInstance
MERGE|AllocateChecklistTask|Apply for credit card|Vision Enterprise Onboarding Step|
ORA_ONB_ENT_ONBOARDING_STEP|United States|1|7109026|N|INI|8153757|8153757|ORA_CHK_MANUAL|ORA_ADHOC_USER|
ORA_CHK_ADHOC_USER|8153756|Vision Enterprise Onboarding|ORA_ONB_ENT_ONBOARDING|United States|1
```

Creating Other Category Allocated Checklists

This example AllocateChecklist.dat file allocates other category checklists using user keys.

```
METADATA|AllocateChecklist|ChecklistName|ChecklistCategory|Country|ChecklistInstance|ChecklistStatus|
PersonNumber|AllowAutoAllocation|InitiatorPersonNumber
MERGE|AllocateChecklist|Envision Offboard|OFFBOARD|United States|1|INI|7109026|N|8153756
METADATA|AllocateChecklistTask|TaskName|ChecklistName|ChecklistCategory|Country|ChecklistInstance|
PersonNumber|MandatoryFlag|Status|TaskPerformerPersonNumber|TaskOwnerPersonNumber|ActionType|PerformerRole|
OwnerRole|InitiatorPersonNumber
MERGE|AllocateChecklistTask|Submit Badge to Admin|Envision Offboard|OFFBOARD|United States|1|7109026|N|INI|
8153757|8153757|ORA_CHK_MANUAL|ORA_ADHOC_USER|ORA_CHK_ADHOC_USER|8153756
```

Note: The Checklist Mass Allocation object doesn't allocate a checklist. The object is used to create the population for mass assigning journeys and allocating scheduled journeys. Mass assigning of journeys currently supports only the person, team, or organization assignment. However, if a user has a custom population, they can use the Checklist Mass Allocation object to create the population and then run the Allocate Scheduled Journeys or Mass Assign Journeys ESS process to actually allocate the journey.

Adding a Task to an Allocated Checklist

This example AllocatedChecklist.dat file creates a task in an allocated checklist.

```
METADATA|AllocateChecklistTask|TaskName|ChecklistName|LegislationCode|ChecklistCategory|ChecklistInstance|
PersonNumber|Description|MandatoryFlag|Status|TaskPerformerPersonNumber|TaskOwnerPersonNumber|ActionType|
PerformerRole|OwnerRole
MERGE|AllocateChecklistTask|Apply for credit card|Onboarding Checklist|US|ONBOARD|1|2304617|Apply for a
 corporate credit card|Y|INI|2304644|2308991|ORA_CHK_MANUAL|ORA_WORKER|ORA_CHK_INITIATOR
```

Modifying a Task in an Allocated Checklist

This example AllocatedChecklist.dat file updates task attributes such as the status, performer, and owner of an allocated task. Updating the task performer using HDL is similar to reassigning the task to another person in the user interface.

```
METADATA|AllocateChecklistTask|TaskName|ChecklistName|Country|ChecklistCategory|ChecklistInstance|
PersonNumber|Description|MandatoryFlag|Status|TaskPerformerPersonNumber|TaskOwnerPersonNumber|ActionType|
PerformerRole|OwnerRole
MERGE|AllocateChecklistTask|Apply for credit card|Onboarding Checklist|United States|ONBOARD|1|2304617|Apply
 for a corporate credit card|Y|SUS|2304644|2308991|ORA_CHK_MANUAL|ORA_WORKER|ORA_CHK_INITIATOR
```

Setting Display Properties for an Allocated Checklist and Allocated Task

This example AllocateChecklist.dat file sets the display properties for an allocated checklist and allocated task.

```
METADATA|AllocateChecklist|ChecklistName|ChecklistCategory|ChecklistInstance|PersonNumber|
InitiatorPersonNumber|ChecklistStatus|AllowAutoAllocation
MERGE|AllocateChecklist|CHK_Test_HDL|ONBOARD|1|8153757|8153756|INI|N
```

```
METADATA|AllocateChecklistTask|TaskName|ChecklistName|ChecklistCategory|ChecklistInstance|PersonNumber|
MandatoryFlag|Status|ActionType
MERGE|AllocateChecklistTask|Collect laptop|CHK_Test_HDL|ONBOARD|1|8153757|Y|INI|ORA_CHK_MANUAL

METADATA|AllocateChecklistProperty|ChecklistName|ChecklistCategory|ChecklistInstance|PersonNumber|Property|
ActionType|ActionSubType|Manager|Assignee|Other|Performer|Owner|PropertyValue
MERGE|AllocateChecklistProperty|CHK_Test_HDL|ONBOARD|1|8153757|ORA_COMPLETED_TASKS|||Hide|Hide|Hide|||
MERGE|AllocateChecklistProperty|CHK_Test_HDL|ONBOARD|1|8153757|ORA_OTHER_INCOMPLETE_TASKS|||Hide|Hide|
Hide|||
MERGE|AllocateChecklistProperty|CHK_Test_HDL|ONBOARD|1|8153757|ORA_ACTION_REMOVE|ORA_CHK_EXTERNAL_URL||
Hide||Hide|Show|Show|
MERGE|AllocateChecklistProperty|CHK_Test_HDL|ONBOARD|1|8153757|ORA_ACTION_REMOVE|ORA_CHK_ESIGN|
ORA_BASIC_SIGN|Hide||Hide|Show|Show|

METADATA|AllocateChecklistTaskProperty|ChecklistName|ChecklistCategory|ChecklistInstance|PersonNumber|
Property|TaskName|Manager|Other|Performer|Owner
MERGE|AllocateChecklistTaskProperty|CHK_Test_HDL|ONBOARD|1|8153757|ORA_ACTION_REASSIGN|Collect laptop|Show|
Show|Show|Show
MERGE|AllocateChecklistTaskProperty|CHK_Test_HDL|ONBOARD|1|8153757|ORA_ACTION_EDIT_DUE_DATE|Collect laptop|
Show|Show|Hide|Hide
```

Deleting a Task in an Allocated Checklist

This example AllocatedTask.dat file deletes a task in an allocated checklist.

```
METADATA|AllocateChecklistTask|TaskName|ChecklistName|ChecklistCategory|ChecklistInstance|PersonNumber
DELETE|AllocateChecklistTask|Apply for credit card|Onboarding Checklist|ONBOARD|1|8153757
```

You can delete an allocated checklist. When you delete an allocated checklist, all associated components are also deleted. This example AllocateChecklist.dat file deletes an allocated checklist and all its associated objects.

```
METADATA|AllocateChecklist|ChecklistName|ChecklistCategory|ChecklistInstance|PersonNumber
DELETE|AllocateChecklist|Onboarding Checklist|ONBOARD|1|8153757
```

Mass Allocate Checklist and Generate Notification

You can use the **Checklist Mass Allocation** HDL business object to mass allocate a checklist for employees and generate a notification for the same.

This example ChecklistMassAllocation.dat file creates a mass allocation object with the list of employees.

```
METADATA|ChecklistMassAllocation|Name|Description|ChecklistAllocationCode|ChecklistName|
InitiatorPersonNumber|Comments|Status|AllocationType|StartDate
MERGE|ChecklistMassAllocation|HDL Mass Allocation|HDL Mass Allocation|HDL_MASS_ALLOC_VISION_CORP|Annual
 review journey|6145367|mass assign journey to all employees in Vision Corporation|ORA_CHK_ACTIVE|ADHOC|
2024/01/10
```

This example ChecklistMassAllocationCriteria.dat file creates a mass allocation criteria object with the list of employees.

```
METADATA|ChecklistMassAllocationCriteria|ChecklistAllocationCode|ChecklistAllocCriteriaCode|ExcludeFlag|
CriteriaType|PersonNumber|AssignmentNumber|EnabledFlag
MERGE|ChecklistMassAllocationCriteria|HDL_MASS_ALLOC_VISION_CORP|HDL2|N|ORA_PERSON|8153756|EEEE8153756|Y
MERGE|ChecklistMassAllocationCriteria|HDL_MASS_ALLOC_VISION_CORP|HDL2|N|ORA_PERSON|9153342|EEEE9153342|Y
MERGE|ChecklistMassAllocationCriteria|HDL_MASS_ALLOC_VISION_CORP|HDL2|N|ORA_PERSON|9472415|EEEE9472415|Y
```

After the objects are created, you need to navigate to Tools > Scheduled Processes and run the Mass Assign Journey process by using the ChecklistMassAllocation identifier used in the ChecklistMassAllocation.dat file. In this example, the HDL_MASS_ALLOC_VISION_CORP identifier is used.

Related Topics

- Guidelines for Loading Checklist Templates
- Guidelines for Loading Allocated Checklists

Guidelines for Loading Areas of Responsibility

Areas of responsibility are responsibilities that a worker has as part of his or her job. Such responsibilities have a defined scope, such as a country or department. For example, a worker may be:

- The Human Resources Representative for a group of organizations
- The Benefits Representative for workers with a specified job
- The Union Representative for workers in Germany

Areas of responsibility are a recommended way of securing access to person records. This topic describes how to load areas of responsibility for a person using HCM Data Loader. Use the **AreasOfResponsibility.dat** file.

Responsibility Type

Responsibility types must exist in the target environment before you can load areas of responsibility. These five responsibility types are available by default, but you can add your own to suit business needs:

- Benefits Representative
- Human Resources Representative
- Payroll Representative
- Recruiting
- Union Representative

The responsibility type is validated against the lookup type PER_RESPONSIBILITY_TYPES.

Responsibility Name

The responsibility name must be unique for the combination of person and responsibility type.

Scope of Responsibility

Various attributes available under scope of responsibility, such as business unit, legal employer, department, and location, must exist before you can load areas of responsibility. For example, a location must exist before you can identify it as the scope of an area of responsibility.

Deleting Areas of Responsibility

You can delete areas of responsibility using HCM Data Loader. However, ensure that deleting an area of responsibility doesn't have unintended consequences. For example, a worker's secured access to person records may be lost if you delete the area of responsibility on which it's based.

Related Topics

- How You Assign Areas of Responsibility
- Guidelines for Securing Person Records

Guidelines for Loading Document Records

Document records store information about documents, such as visas, licenses, and medical certificates, for a person. Document records can include electronic versions of the documents as attachments.

This topic describes how to load Document Record and Document Record Attachment components for a person using HCM Data Loader.

Document Type

Document records exist for a specific document type. You must ensure that the document type exists in the target environment. The definition of the document type identifies supported and required attributes.

Attachment Files

The Document Record Attachment component holds the electronic version of a document for a person. Use the attribute **URLorTextorFilename** to supply the:

- URL, if the DataTypeCode is WEB_PAGE
- Text, if the DataTypeCode is TEXT
- File name, if the DataTypeCode is FILE

If you're uploading attachment files, then they must be in the same .zip file as the DocumentsOfRecord.dat file that contains the related document records. All attachment files must be placed in a subdirectory named **BlobFiles**. This subdirectory is named for the BLOB data-type format in which attachment files are held. You specify the name of the attachment file on the **File** attribute of the Document Record Attachment component.

You can overwrite an existing attachment for a document record with a new attachment using HCM Data Loader. However, the overwriting happens only if the name of the existing attachment and the new attachment are the same.

> **Note:** You can't overwrite a document record attachment on the Document Details page. The UI still requires you to first delete the existing attachment file and then upload the new file.

Loading Document Records

You supply document record data in a DocumentsOfRecord.dat file. This example DocumentsOfRecord.dat file creates a passport document for a person and includes a PDF file of the passport as an attachment. It identifies both the document record and the attachment using source keys.

```
METADATA|DocumentsOfRecord|PersonNumber|DocumentType|Country|DocumentCode|DocumentName|DocumentNumber|
AssignmentNumber|DateFrom|DateTo|IssuingAuthority|IssuedDate|IssuingCountryName|IssuingLocation|Comments
MERGE|DocumentsOfRecord|134003891|Passport|United Kingdom|PASSPORT_2442|UK Passport|355429|PS098123|
2012/04/03|2022/04/03|United Kingdom Passport Control|2012/04/03|UK|London|JsmithPassportUpload
METADATA|DocumentAttachment|PersonNumber|DocumentCode|DataTypeCode|URLorTextorFileName|Title|File
```

```
MERGE|DocumentAttachment|134003891|PASSPORT_2442|FILE|Passport_2442|JSmithPassport|JSmithPassport.pdf
```

Deleting Document Records

You can delete Document Record objects using HCM Data Loader. When you delete a document record, any associated attachment record is deleted automatically. You can delete just the attachment, if appropriate.

This example DocumentsOfRecord.dat file deletes an existing document record. It identifies the document record using its source key.

```
METADATA|DocumentsOfRecord|DocumentdOfRecordId
DELETE|DocumentsOfRecord|1234567890
```

Related Topics

- How You Load Images, Attachments, and Large Strings

Guidelines for Loading Time Record Groups

A time record group includes related time entries, such as all reported time entries in a weekly time card. You can use HCM Data Loader to load only approved time record groups, meaning historical data. You can't edit time record groups after upload.

This topic describes aspects of the Time Record Group object that you must understand to load time record groups successfully using HCM Data Loader.

Preparing to Load Time Record Groups

Before you can load time record groups:

- Setup of Oracle Fusion Time and Labor must be complete.
- Relevant person records, each with a valid worker assignment, must exist in the target environment.
- A worker time processing profile must exist for each worker whose time card details you're loading. The time processing profile may identify other consumers of the time card data.
    - If Oracle Fusion Global Payroll is a consumer of time card data, then the relevant payroll element entries must exist.
    - If Oracle Fusion Project Costing is a consumer of time card data, then the relevant Project Costing setup data must exist.

You may also need:

- Time rules, if the data being loaded required validation and calculation
- A worker time entry profile, if the worker or time and labor manager can view the loaded data in the application

Supplying Keys

The Time Record Group object doesn't support source keys. Therefore, supply user keys when creating time record groups.

Querying Predefined Time Attributes

Predefined time attributes and their data types aren't visible on a user interface. To query time attribute names, use this SQL query:

```
SELECT ATRB.NAME, ATRB.DISPLAY_NAME, ATRB.CLASS, ATRB.ATTRIBUTE_TYPE
FROM HWM_TM_ATRB_FLDS_VL ATRB
WHERE ATRB.ATTRIBUTE_CATEGORY NOT IN ('TIME_BUILDING_BLOCK', 'TIME_RECORD',
'TIME_RECORD_GROUP');
```

This table shows example results from this query.

Attribute Name	Attribute Display Name	Attribute Class	Attribute Data Type
PJC_PROJECT_ID	Project	SIMPLE	Number
PayrollTimeType	Payroll Time Type	MASTER	Varchar

Updating Time Record Groups

When updating a time record group, you must supply the complete Time Record Group object in the TimeRecordGroup.dat file, not just the changed records.

Deleting Time Record Groups

You can delete Time Record Group objects using HCM Data Loader only if you identify them using surrogate IDs. When you delete a Time Record Group object, its child components are also deleted. You can't delete individual components of the object. This example TimeRecordGroup.dat file deletes a time record group and its child components. It references the record by its surrogate ID.

```
METADATA|TimeRecordGroup|TimeRecordGroupId|ResourceId|ResourceType|TcStartTime|TcStopTime|GroupType|
SubResourceId
DELETE|TimeRecordGroup|100000001|100000008153756|PERSON|2010/01/04 00:00:01|2010/01/09 23:59:59|RPTD_TIME|
100000008154060
```

To obtain the surrogate IDs of existing objects, use the HCM Data Loader extract Integration Object User Key Map Extract.

Related Topics

- Examples of Loading Time Record Groups
- Overview of HCM Data Loader Extracts

Examples of Loading Time Record Groups

This topic shows how to create Time Record Group objects using HCM Data Loader.

Creating Time Record Groups

This example TimeRecordGroup.dat file creates two time record groups for the same assignment. It references the person and assignment by their user keys.

```
METADATA|TimeRecordGroup|GroupType|ResourceType|PersonNumber|AssignmentNumber|TcStartTime|TcStopTime
MERGE|TimeRecordGroup|RPTD_TIME|PERSON|8153756|EEEE8153756|2010/02/01 00:00:01|2010/02/06 23:59:59
METADATA|TimeRecord|GroupType|ResourceType|PersonNumber|TcStartTime|TcStopTime|OrderEntered|StartTime|
StopTime|TmRecType|AssignmentNumber|Measure|UnitOfMeasure
MERGE|TimeRecord|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|1|2010/02/01 09:00:01|
2010/02/01 13:00:01|RANGE|EEEE8153756|4|HR
MERGE|TimeRecord|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|2|2010/02/02 09:00:01|
2010/02/02 13:00:01|RANGE|EEEE8153756|4|HR
MERGE|TimeRecord|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|3|2010/02/03 09:00:01|
2010/02/03 13:00:01|RANGE|EEEE8153756|4|HR
MERGE|TimeRecord|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|4|2010/02/04 09:00:01|
2010/02/04 13:00:01|RANGE|EEEE8153756|4|HR
MERGE|TimeRecord|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|5|2010/02/05 09:00:01|
2010/02/05 13:00:01|RANGE|EEEE8153756|4|HR
METADATA|TimeRecordGroupAttribute|GroupType|ResourceType|PersonNumber|TcStartTime|TcStopTime|AttributeName|
AttributeDataType|AttributeStringValue
MERGE|TimeRecordGroupAttribute|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|Comment|
String|Create Timecard for 2 week of year 2010
METADATA|TimeRepositoryAttribute|GroupType|ResourceType|PersonNumber|TcStartTime|TcStopTime|OrderEntered|
AttributeName|AttributeDataType|AttributeStringValue|AttributeBigDecimalValue
MERGE|TimeRepositoryAttribute|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|1|
PayrollTimeType|STRING|ZOTL_Regular|
MERGE|TimeRepositoryAttribute|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|1|LDG_ID|
BIG_DECIMAL||202
MERGE|TimeRepositoryAttribute|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|2|
PayrollTimeType|STRING|ZOTL_Regular|
MERGE|TimeRepositoryAttribute|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|2|LDG_ID|
BIG_DECIMAL||202
MERGE|TimeRepositoryAttribute|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|3|
PayrollTimeType|STRING|ZOTL_Regular|
MERGE|TimeRepositoryAttribute|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|3|LDG_ID|
BIG_DECIMAL||202
MERGE|TimeRepositoryAttribute|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|4|
PayrollTimeType|STRING|ZOTL_Regular|
MERGE|TimeRepositoryAttribute|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|4|LDG_ID|
BIG_DECIMAL||202
MERGE|TimeRepositoryAttribute|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|5|
PayrollTimeType|STRING|ZOTL_Regular|
MERGE|TimeRepositoryAttribute|RPTD_TIME|PERSON|8153756|2010/02/01 00:00:01|2010/02/06 23:59:59|5|LDG_ID|
BIG_DECIMAL||202
```

This example demonstrates that:

- The **GroupType** attribute value on all components must be **RPTD_TIME**.
- The **ResourceType** attribute value on all components must be **PERSON**.
- The **TcStartTime** and **TcStopTime** attribute values on all child components of the object must match those on the parent Time Record Group object.
- You must supply the same value for the **OrderEntered** attribute on both the Time Repository Attribute component and its parent Time Record component.

Related Topics

- Guidelines for Loading Time Record Groups

Guidelines for Loading User-Update Requests

You use the User object to create requests to update existing Oracle HCM Cloud user accounts. This topic describes when to use each component of the User object. It also describes any restrictions on its use.

The User Component

You can use the User component to create requests to:

- Update a user name.
- Suspend or activate a user.
- Add a single role to or remove a single role from a user.
- Create a user account, in specific circumstances.
- Delete a user account.

You also use this component to update the **CredentialsEmailSent** value in the PER_USERS table. This value records whether the account and password details have been sent to the user. Valid values are **Y** and **N**. After the email has been sent, it can't be sent again unless you reset this value. For example, to reset user passwords, you can run the **Send User Name and Password Email Notifications** process. Before doing so, you can set **CredentialsEmailSent** to **N** to ensure that users are notified of new passwords. Otherwise, they aren't notified.

The User Role Component

The User Role component is a child of the User object. You use the User Role component to add or remove multiple roles for a user. To add or remove a single role, use the User component.

Creating User Accounts

Depending on the setting of the **User Account Creation** enterprise option, user accounts can be created automatically for workers loaded in bulk. However, you may have prevented the user account from being created for an individual worker. Alternatively, it may have failed to be created for some reason. In these cases, you can use the User object to request a user account.

> **Note:** Use the User object to request a user account only as an exception. The standard method is to request the user account when you create the worker.

A user account can be created by means of the User object only if the **User Account Creation** enterprise option enables automatic creation of user accounts.

Deleting User Accounts

On occasion, you may want to delete a user account. For example, you may have created the user name in the wrong format. You can delete user accounts in both test and production environments, and the accounts can be in any status. The User and Role Provisioning options have no effect on user-account deletion. For example, deletion requests can't be suppressed.

Tip: The user is deleted from both the LDAP directory and the PER_USERS, PER_USER_ROLES, and PER_USER_HISTORY records. Therefore, you can reuse the user name. You can recreate the user account on the Security Console, using the **Manage Users** task, or by loading the User object.

Running Send Pending LDAP Requests

After loading user-update requests, you run the **Send Pending LDAP Requests** process to deliver them to your LDAP directory server. The recommendation is to schedule this process to run daily. You don't have to run this process after updating the **CredentialsEmailSent** indicator.

Related Topics

- Examples of Loading User-Update Requests
- Why You Should Run the Send Pending LDAP Requests Process

Examples of Loading User-Update Requests

Use the User object to manage existing users and their roles. This topic provides examples showing how to create some typical user-update requests.

Updating a User

This example creates a request to update the user name for the user identified by the specified person number.

```
METADATA|User|PersonNumber|Username
MERGE|User|12312|carlton.baugh@vision.com
```

This example creates a request to suspend an active user.

```
METADATA|User|PersonNumber|Suspended
MERGE|User|12312|Y
```

This example creates a request to activate a suspended user.

```
METADATA|User|PersonNumber|Suspended
MERGE|User|12312|N
```

This example updates the **CredentialsEmailSent** indicator, which determines whether an email containing the user credentials is sent to the user. If the **CredentialsEmailSent** indicator is **Y**, then no email is sent. If the indicator is **N**, then an email is sent. In this example, the indicator is changed from **Y** to **N**.

```
METADATA|User|PersonNumber|CredentialsEmailSent
MERGE|User|12312|N
```

Updating User Roles

You use the User component to add a single role. Note that you supply role codes rather than role names. For example, you specify ORA_PER_EMPLOYEE_ABSTRACT rather than Employee to add the employee role. If you supply an invalid role code, then the request is created but any attempt to provision the role fails.

This example creates a request to add a single role to the user.

```
METADATA|User|PersonNumber|RoleCommonName|AddRemoveRole
```

```
MERGE|User|12312|ORA_PER_EMPLOYEE_ABSTRACT|ADD
```

This example creates a request to add multiple roles to the user. You use the User Role component to add multiple roles.

```
METADATA|User|PersonNumber
MERGE|User|12312
METADATA|UserRole|PersonNumber|RoleCommonName|AddRemoveRole
MERGE|UserRole|12312|ORA_PER_EMPLOYEE_ABSTRACT|ADD
MERGE|UserRole|12312|ORA_PER_LINE_MANAGER_ABSTRACT|ADD
```

This example creates a request to remove a single role from the user. You use the User component to remove a single role.

```
METADATA|User|PersonNumber|RoleCommonName|AddRemoveRole
MERGE|User|12312|ORA_PER_EMPLOYEE_ABSTRACT|REMOVE
```

This example creates a request to remove multiple roles from the user. You use the User Role component to remove multiple roles.

```
METADATA|User|PersonNumber
MERGE|User|12312
METADATA|UserRole|PersonNumber|RoleCommonName|AddRemoveRole
MERGE|UserRole|12312|ORA_PER_EMPLOYEE_ABSTRACT|REMOVE
MERGE|UserRole|12312|ORA_PER_LINE_MANAGER_ABSTRACT|REMOVE
```

Creating a User Account

You can use the User object to create a user account in limited circumstances. For example:

- An attempt to request a user account when loading the Worker object has failed.
- The **GeneratedUserAccountFlag** attribute of the Person User Information component was set to **N** either by mistake or intentionally for the loaded worker.

Creating a user account is a two-step process:

1. Create the user.
2. Activate the user and add roles.

This example creates a request to create a user account for the specified person number. You can also supply a user name if required. If you supply no user name, then the user name is generated in the default format as specified on the Security Console. Once the account exists, it's suspended immediately as it has no roles.

```
METADATA|User|PersonNumber|GenerateUserAccount
MERGE|User|12312|Y
```

This example activates the user account and adds roles so that the account remains active.

```
METADATA|User|PersonNumber|Suspended
METADATA|UserRole|PersonNumber|RoleCommonName|AddRemoveRole
MERGE|User|12312|N
MERGE|UserRole|12312|ORA_PER_EMPLOYEE_ABSTRACT|ADD
MERGE|UserRole|12312|ORA_PER_LINE_MANAGER_ABSTRACT|ADD
```

> **Tip:** User-update requests aren't processed until **Send Pending LDAP Requests** next runs. If the process isn't scheduled, remember to run it.

Deleting a User Account

This example deletes the user account for the worker with the specified person number.

```
METADATA|User|PersonNumber
DELETE|User|12312
```

Related Topics

- Guidelines for Loading User-Update Requests

Guidelines for Loading Worker Schedules

A worker schedule defines a worker's shift pattern for a specified period. To load worker schedules using HCM Data Loader, use the Schedule Request object. This topic describes some aspects of worker schedules that you must understand to load them successfully.

Preparing to Load Worker Schedules

Before you load worker schedules, you must define shift codes and scheduler profiles. You can also define shift layouts, if required. This table identifies the relevant tasks. Perform these tasks in the Time Management work area.

Task	Description
Manage Shift Properties	Defines shift codes for all shifts
Manage Scheduler Profiles	Defines scheduler profiles. Ensure that each of the workers whose shifts you load appears in a scheduler profile.
Manage Layout Sets	Defines a shift layout for loaded time attributes. This task is optional. If you don't define the shift layout, then you can still load time attributes but you can't see them in the application.

You can also set up and enable notifications for schedulers and workers. This table identifies the notifications. Configure these notifications in the Alerts Composer work area.

Notification	Description
HTS Worker Shifts Imported	Sent to managers and schedulers when a new schedule is loaded
HTS Schedule Publication	Sent to workers when a new schedule is published

Tip: You can also enable or disable worker notifications by setting the **WorkerNotification** attribute of the Schedule Event component when you load schedule requests.

How Worker Schedules Are Loaded

Unlike most other HCM Data Loader business objects, worker schedules aren't loaded from the HCM Data Loader stage tables to the application tables. Instead, worker schedules are loaded from the HCM Data Loader stage tables to the

schedule stage tables. You must then run the **Process Imported Shifts** process to load worker schedules into either the planned or the planned and published schedule tables. You run the **Process Imported Shifts** process using the **Manage Scheduled Processes** task in the Time Management work area. This process:

- Merges all schedule events for a worker from HCM Data Loader and REST services
- Consolidates the schedule events for a worker from multiple schedule requests, orders them by request time, and processes them in this order

This approach to loading Schedule Request objects maintains data integrity. However, you see only basic validation errors from HCM Data Loader. Most validation is performed when **Process Imported Shifts** loads Schedule Request objects into the schedule tables.

Import Modes

When loading Schedule Request objects, you must set the **ImportMode** attribute of the Schedule Event component to either **FULL** or **UPDATE**. This table describes the import modes.

Import Mode	Description
FULL	Use **FULL** mode when you create worker schedules. For example, use this mode to load worker schedules during implementation. When the mode is **FULL**, you must set the **ShiftAction** attribute of all Schedule Shift Event components in the Schedule Event to **CREATE**.
UPDATE	Use **UPDATE** mode to update or delete existing worker schedules. When the mode is **UPDATE**, you can create additional shifts, update existing shifts, or delete existing shifts. You set the **ShiftAction** attribute of the Schedule Shift Event component as appropriate.

Verifying Loaded Schedules

Time and labor managers and schedulers verify loaded schedules in the Time Management work area. They verify:

- Planned schedules on the Manage Planned Schedule page
- Published schedules using the **View Published Schedule** task

Workers view their published schedules using the calendar and Team Schedule pages of the Time work area.

Related Topics

- Examples of Creating Worker Schedules
- Examples of Updating Worker Schedules

Examples of Creating Worker Schedules

This topic provides examples showing how to create worker schedules using HCM Data Loader.

Creating a Schedule for a Worker for a Specific Week

For each worker whose schedule you're creating, you must load Schedule Request, Schedule Event, and Schedule Shift Event components. On the Schedule Event component, you set the **ImportMode** attribute to **FULL**. On the Schedule Shift Event components, you set the **ShiftAction** attribute to **CREATE**.

This example ScheduleRequest.dat file creates a schedule for a worker for a specified week. It uses user keys.

```
METADATA|ScheduleRequest|ScheduleRequestNumber|RequestSource|RequestTime
MERGE|ScheduleRequest|SR1234501|3RD_PARTY_ABC|2017-01-01T13:25:20.010+01:00
METADATA|ScheduleEvent|ScheduleRequestNumber|ScheduleEventNumber|ImportMode|PersonNumber|PeriodStartDate|
PeriodEndDate|AllowEdits|Publish|WorkerNotification
MERGE|ScheduleEvent|SR1234501|SE001|FULL|955160008182092|2017/01/01|2017/01/07|Y|Y|N
METADATA|ScheduleShiftEvent|ScheduleRequestNumber|ScheduleEventNumber|ScheduleShiftEventNumber|ShiftAction|
ReferenceDay|ShiftStartTime|ShiftEndTime|ShiftTimeNotWorked|ShiftCode|ShiftCategory|ShiftType|AllowEdits
MERGE|ScheduleShiftEvent|SR1234501|SE001|SSE101|CREATE|2017/01/01|2017-01-01T09:00:00+01:00|
2017-01-01T17:00:00+01:00|60|MORNTIME|WORK|TIME|N
MERGE|ScheduleShiftEvent|SR1234501|SE001|SSE102|CREATE|2017/01/02|2017-01-02T09:00:00+01:00|
2017-01-02T17:00:00+01:00|60|MORNTIME|WORK|TIME|N
MERGE|ScheduleShiftEvent|SR1234501|SE001|SSE103|CREATE|2017/01/03|2017-01-03T09:00:00+01:00|
2017-04-03T17:00:00+01:00|60|MORNTIME|WORK|TIME|N
MERGE|ScheduleShiftEvent|SR1234501|SE001|SSE104|CREATE|2017/01/04|2017-01-04T09:00:00+01:00|
2017-04-04T17:00:00+01:00|60|MORNTIME|WORK|TIME|N
MERGE|ScheduleShiftEvent|SR1234501|SE001|SSE105|CREATE|2017/01/05|2017-01-05T09:00:00+01:00|
2017-04-05T17:00:00+01:00|60|MORNTIME|WORK|TIME|N
METADATA|ScheduleShiftAttribute|ScheduleRequestNumber|ScheduleEventNumber|ScheduleShiftEventNumber|
ScheduleShiftAttributeNumber|AttributeName|AttributeValue
MERGE|ScheduleShiftAttribute|SR1234501|SE001|SSE101|SSA001|PayrollTimeType|WFM_PAY_REGULAR_US
MERGE|ScheduleShiftAttribute|SR1234501|SE001|SSE101|SSA002|GD_Department_CHAR|1000
MERGE|ScheduleShiftAttribute|SR1234501|SE001|SSE102|SSA001|PayrollTimeType|WFM_PAY_REGULAR_US
MERGE|ScheduleShiftAttribute|SR1234501|SE001|SSE102|SSA002|GD_Department_CHAR|1000
MERGE|ScheduleShiftAttribute|SR1234501|SE001|SSE103|SSA001|PayrollTimeType|WFM_PAY_REGULAR_US
MERGE|ScheduleShiftAttribute|SR1234501|SE001|SSE103|SSA002|GD_Department_CHAR|1000
MERGE|ScheduleShiftAttribute|SR1234501|SE001|SSE104|SSA001|PayrollTimeType|WFM_PAY_REGULAR_US
MERGE|ScheduleShiftAttribute|SR1234501|SE001|SSE104|SSA002|GD_Department_CHAR|1000
MERGE|ScheduleShiftAttribute|SR1234501|SE001|SSE105|SSA001|PayrollTimeType|WFM_PAY_REGULAR_US
MERGE|ScheduleShiftAttribute|SR1234501|SE001|SSE105|SSA002|GD_Department_CHAR|1000
```

Creating a Shift for a Worker for a Specific Day

To create a single shift for a worker, you load the Schedule Request, Schedule Event, and Schedule Shift Event components. On the Schedule Event component, you set the **ImportMode** attribute to **FULL**. On the Schedule Shift Event component, you set the **ShiftAction** attribute to **CREATE**.

This example ScheduleRequest.dat file creates a shift for a worker on a specific day. It uses user keys.

```
METADATA|ScheduleRequest|ScheduleRequestNumber|RequestSource|RequestTime
MERGE|ScheduleRequest|SR1234502|3RD_PARTY_ABC|2017-01-01T13:25:20.010+01:00
METADATA|ScheduleEvent|ScheduleRequestNumber|ScheduleEventNumber|ImportMode|PersonNumber|PeriodStartDate|
PeriodEndDate|AllowEdits|Publish|WorkerNotification
MERGE|ScheduleEvent|SR1234502|SE001|FULL|955160008272091|2017/01/01|2017/01/07|Y|Y|N
METADATA|ScheduleShiftEvent|ScheduleRequestNumber|ScheduleEventNumber|ScheduleShiftEventNumber|ShiftNumber|
ShiftAction|ReferenceDay|ShiftStartTime|ShiftEndTime|ShiftDuration|ShiftTimeNotWorked|ShiftCode|
ShiftCategory|ShiftType|AllowEdits
MERGE|ScheduleShiftEvent|SR1234502|SE001|SSE101|SN001|CREATE|2017/01/01|2017-01-01T06:00:00+01:00|
2017-01-01T09:00:00+01:00||60|MORNTIME|WORK|TIME|N
METADATA|ScheduleShiftAttribute|ScheduleRequestNumber|ScheduleEventNumber|ScheduleShiftEventNumber|
ScheduleShiftAttributeNumber|AttributeName|AttributeValue
MERGE|ScheduleShiftAttribute|SR1234502|SE001|SSE101|SSA001|PayrollTimeType|WFM_PAY_REGULAR_US
MERGE|ScheduleShiftAttribute|SR1234502|SE001|SSE101|SSA002|GD_Department_CHAR|1000
```

Note: Shift numbers are optional when you create a schedule. Typically, they're generated by a third-party scheduling system. You must specify a shift number when updating or deleting a shift.

Related Topics

- Guidelines for Loading Worker Schedules
- Examples of Updating Worker Schedules

Examples of Updating Worker Schedules

This topic provides examples showing how to update existing worker schedules using HCM Data Loader.

Updating a Shift

To update an existing shift, you load the Schedule Request, Schedule Event, and Schedule Shift Event components. On the Schedule Event component, you set the **ImportMode** attribute to **UPDATE**. On the Schedule Shift Event component, you specify the shift number and set the **ShiftAction** attribute to **UPDATE**.

This example ScheduleRequest.dat file updates an existing worker shift from a third-party application.

```
METADATA|ScheduleRequest|ScheduleRequestNumber|RequestSource|RequestTime
MERGE|ScheduleRequest|SR1234503|3RD_PARTY_ABC|2017-01-01T13:25:20.010+01:00
METADATA|ScheduleEvent|ScheduleRequestNumber|ScheduleEventNumber|ImportMode|PersonNumber|PeriodStartDate|
PeriodEndDate|AllowEdits|Publish|WorkerNotification
MERGE|ScheduleEvent|SR1234503|SE001|UPDATE|955160008272091|2017/01/01|2017/01/07|Y|Y|N
METADATA|ScheduleShiftEvent|ScheduleRequestNumber|ScheduleEventNumber|ScheduleShiftEventNumber|ShiftNumber|
ShiftAction|ReferenceDay|ShiftStartTime|ShiftEndTime|ShiftTimeNotWorked|ShiftCode|ShiftCategory|ShiftType|
AllowEdits
MERGE|ScheduleShiftEvent|SR1234503|SE001|SSE101|SN001|UPDATE|2017/01/01|2017-01-01T07:00:00+01:00|
2017-01-01T11:00:00+01:00|60|MORNTIME|WORK|TIME|N
```

Deleting a Shift

To delete a shift, you load the Schedule Request, Schedule Event, and Schedule Shift Event components. On the Schedule Event component, you set the **ImportMode** attribute to **UPDATE**. On the Schedule Shift Event component, you specify the shift number and set the **ShiftAction** attribute to **DELETE**.

This example ScheduleRequest.dat file deletes an existing shift

```
METADATA|ScheduleRequest|ScheduleRequestNumber|RequestSource|RequestTime
MERGE|ScheduleRequest|SR1234504|3RD_PARTY_ABC|2017-01-01T13:25:20.010+01:00
METADATA|ScheduleEvent|ScheduleRequestNumber|ScheduleEventNumber|ImportMode|PersonNumber|PeriodStartDate|
PeriodEndDate|AllowEdits|Publish|WorkerNotification
MERGE|ScheduleEvent|SR1234504|SE001|UPDATE|955160008272091|2017/01/01|2017/01/07|Y|Y|N
METADATA|ScheduleShiftEvent|ScheduleRequestNumber|ScheduleEventNumber|ScheduleShiftEventNumber|ShiftNumber|
ShiftAction|ReferenceDay|ShiftStartTime|ShiftEndTime|ShiftDuration|ShiftTimeNotWorked|ShiftCode|
ShiftCategory|ShiftType|AllowEdits
MERGE|ScheduleShiftEvent|SR1234504|SE001|SSE101|SN001|DELETE|2017/01/01|2017-01-01T07:00:00+01:00|
2017-01-01T11:00:00+01:00||60|MORNTIME|WORK|TIME|N
```

Clearing a Schedule

To clear a worker schedule, load the Schedule Request and Schedule Event components. On the Schedule Event component, you set the **ImportMode** attribute to **FULL**. Because you aren't providing Schedule Shift Event components, the Process Imported Shifts process deletes all shifts for the given period.

This example ScheduleRequest.dat file clears all existing shifts for a worker for a specified period.

```
METADATA|ScheduleRequest|ScheduleRequestNumber|RequestSource|RequestTime
MERGE|ScheduleRequest|SR1234537|3RD_PARTY_ABC|2017-05-01T13:25:20.010+01:00
METADATA|ScheduleEvent|ScheduleRequestNumber|ScheduleEventNumber|ImportMode|PersonNumber|PeriodStartDate|
PeriodEndDate|AllowEdits|Publish|WorkerNotification
MERGE|ScheduleEvent|SR1234537|SE002|FULL|955160008182092|2017/05/01|2017/05/31|Y|Y|N
```

Related Topics

- Guidelines for Loading Worker Schedules
- Examples of Creating Worker Schedules

Guidelines for Loading Skills

Create skills that workers can select and use in their personal brand profile. This topic describes how to load skills, using HCM Data Loader.

Skill Types

Worker skill types can be either interpersonal or specialized. Skill types must exist in the target environment before you load skills.

Loading Skills

You supply the skill in the **BrandSkill.dat** file. This example **BrandSkill.dat** file creates a specialized skill namely Kubernetes.

```
METADATA|BrandSkill|SkillType|SkillCode|SkillName|SourceSystemOwner|SourceSystemId
MERGE|BrandSkill|SPECIALIZED|kubernetes|Kubernetes|HWR|HWR_DEVELOPMENT
```

FAQs for Loading Worker-Related Objects

Can I upload documents using AllocateChecklist.dat?

No, you can't. Currently, attachments aren't supported in HDL. However, if the checklist is auto allocated using HDL and the setup has some attachments then the reference to those attachments are copied. But you can't attach documents in an already allocated checklist or task.

5 Loading Work Structures

Overview of Loading Work Structures

The work structures that you can load using HCM Data Loader include grades, grade ladders, grade rates, jobs, job families, locations, organizations, positions, and HCM position hierarchies. This topic describes some general considerations that apply to most or all work structures.

Reviewing Component Hierarchies

Select a work structure on the View Business Objects page to review its component hierarchy.

Loading Set-Enabled Work Structures

These work structures are set enabled:

- Departments
- Grades
- Jobs
- Locations

You can create them in a specific reference data set, which restricts their use to an individual business unit. Alternatively, you can create them in the common set, where they're generally available. You must ensure that referenced sets exist and are associated with relevant business units, if appropriate.

Using Action Reasons

You can associate action reasons with these work structures:

- Grades
- Grade rates
- Grade ladders
- Jobs
- Locations
- Organizations
- Positions

If you plan to use action reasons, then you must create them before you load work structures. In the Setup and Maintenance work area, use the following:

- Functional Area: Workforce Structures
- Task: Manage Action Reasons

Related Topics

- Sources of Business-Object Information

Examples of Loading Collective Agreements

A collective agreement is a written agreement between an employer and a union or bargaining unit. It defines terms and conditions of employment for the represented workers. The terms used to refer to collective agreements may vary by country.

Generally, the agreement is for a specified term. This topic describes aspects of the Collective Agreement object that you must understand to load them successfully using HCM Data Loader.

Preparing to Load Collective Agreements

A collective agreement is country-specific. Therefore, **LegislationCode** is a required attribute of the Collective Agreement object.

You can associate any combination of a union, a bargaining unit, and a legal employer with the collective agreement. For example, you can associate a collective agreement with a union or with both a union and a legal employer. Any union, bargaining unit, or legal employer associated with collective agreements must exist in the target environment before you load those collective agreements.

Creating Collective Agreements

You supply collective agreements data in the CollectiveAgreements.dat file for processing by HCM Data Loader. This example CollectiveAgreements.dat file creates a collective agreement using source keys. The collective agreement is associated with a union, a bargaining unit, and a legal employer.

```
METADATA|CollectiveAgreements|SourceSystemOwner|SourceSystemId|CollectiveAgreementName|UnionName|
EffectiveStartDate|EffectiveEndDate|Status|IdentificationCode|LegalEmployerName|LegislationCode|Description|
BargainingUnitCode|EmployeeOrgName|EmployeeOrgContact|EmployerOrgName|EmployerOrgContact
MERGE|CollectiveAgreements|VISION|VISCAGR01|SEIU UHW Medical Social Workers (California, Northern)|Service
 Employees International Union|2010/01/01|4712/12/31|A|SEIU_UHW_MSW_CA_N|Vision Corporation|US|Collective
 Agreement covering the medical social workers in northern California affiliated to the Service Employees
 International Union.|BU_SEIU_UHW_MSW_CA_N|Vision Corporation|John Gorman|Vision Corporation|Jane Reifer
```

This example CollectiveAgreements.dat file creates a collective agreement using user keys. The collective agreement is associated with a union, a bargaining unit, and a legal employer.

```
METADATA|CollectiveAgreements|CollectiveAgreementName|UnionName|EffectiveStartDate|EffectiveEndDate|Status|
IdentificationCode|LegalEmployerName|LegislationCode|Description|BargainingUnitCode|EmployeeOrgName|
EmployeeOrgContact|EmployerOrgName|EmployerOrgContact
MERGE|CollectiveAgreements|SEIU UHW Medical Social Workers (California, Northern)|Service Employees
 International Union|2010/01/01|4712/12/31|A|SEIU_UHW_MSW_CA_N|Vision Corporation|US|Collective
 Agreement covering the medical social workers in northern California affiliated to the Service Employees
 International Union.|BU_SEIU_UHW_MSW_CA_N|Vision Corporation|John Gorman|Vision Corporation|Jane Reifer
```

You can't load attachments for collective agreements using HCM Data Loader.

Updating Collective Agreements

You can't change either the first effective start date or last effective end date for an existing collective agreement record. Therefore, create your collective agreement records with an effective start date on or before the start date of any record that references them.

If the collective agreement is linked to an assignment, then you can't edit the **IdentificationCode**, **LegislationCode**, **BargainingUnitCode**, **LegalEmployerName**, or **LegalEntityId** attributes.

Loading Translated Collective Agreement Names

This example CollectiveAgreementsTranslation.dat file translates the name of an existing collective agreement. It identifies the existing record using source keys.

```
METADATA|CollectiveAgreementsTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|
EffectiveEndDate|Language|CollectiveAgreementName|Description
MERGE|CollectiveAgreementsTranslation|VISION|VISCAGR01|2001/01/01|4712/12/31|FR|SEIU UHW Travailleurs
 sociaux medicaux (Californie, Nord)|Convention collective couvrant les travailleurs sociaux medicaux dans
 le nord de la Californie affiliee a l'Union internationale des employes du service.
```

Related Topics

- Worker Union Management

Guidelines for Loading Grades

A grade identifies a worker's rank or level of compensation. You can create grades either with or without grade steps. This topic describes aspects of the Grade object that you must understand to load grades successfully.

Grade Steps

If you're using grade steps, then load one Grade Step component for each grade step.

Changing Logical Start Dates for Grades

You can change the first effective start date for an existing grade. However, the new effective start date must be before the existing effective start date. Create Grade objects with effective start dates on or before the start dates of other objects that refer to your grades.

Loading Translated Grade and Grade-Step Names

Supply grade and grade-step names in the language of the user who's loading them. You supply a GradeTranslation.dat file to translate grade names into other enabled languages once the grades exist. You supply a GradeStepTranslation.dat file to translate grade-step names.

This example GradeTranslation.dat file translates the names of existing grades. It identifies the grades by their source keys.

```
METADATA|GradeTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Language|GradeName
MERGE|GradeTranslation|VISION|ADMIN|1951/01/01|FR|Administrateur
MERGE|GradeTranslation|VISION|IC1|1951/01/01|FR|Contributeur individuel
MERGE|GradeTranslation|VISION|M1|1951/01/01|FR|Superviseur
```

```
MERGE|GradeTranslation|VISION|D1|1951/01/01|FR|Directeur
```

This example GradeStepTranslation.dat file translates the names of existing grade steps. It identifies the grade steps by their source keys.

```
METADATA|GradeStepTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Language|GradeStepName
MERGE|GradeStepTranslation|VISION|LT_CDR1|1951/01/01|FR|Niveau 1
MERGE|GradeStepTranslation|VISION|LT_CDR2|1951/01/01|FR|Niveau 2
MERGE|GradeStepTranslation|VISION|LT_CDR3|1951/01/01|FR|Niveau 3
```

Deleting Grades

You can delete a Grade object using HCM Data Loader, provided that it's not referenced by other objects, such as Assignment, Job, or Grade Rate. This example Grade.dat file deletes existing grades and any associated grade steps. It identifies grades by their source keys.

```
METADATA|Grade|SourceSystemOwner|SourceSystemId|EffectiveStartDate
DELETE|Grade|VISION|CADET|1951/01/01
DELETE|Grade|VISION|LT_CDR|1951/01/01
```

Related Topics

- Overview of Loading Work Structures

Guidelines for Loading Grade Ladders

A grade ladder is a sequence of grades, where the grade order represents a worker's progression. This topic describes aspects of the Grade Ladder object that you must understand to load grade ladders successfully.

The Grade-Ladder Component Hierarchy

The grade ladder is made up of these components:

- Grade Ladder.
- Grades in Ladder. Include one component of this type for each grade, regardless of the grade ladder type.
- Step Rate. Include at least one component of this type for grade ladders of type STEP.
- Step Rate Value. Include one component of this type for each step in a grade ladder of type STEP.

Loading Translated Grade-Ladder and Step-Rate Names

Supply grade-ladder and step-rate names in the language of the user who's loading them. You supply a GradeLadderTranslation.dat file to translate grade-ladder names into other enabled languages once the grade ladders exist. You supply a StepRateTranslation.dat file to translate step-rate names. This example GradeLadderTranslation.dat file translates the name of an existing grade ladder. It identifies the grade ladder by its source key.

```
METADATA|GradeLadderTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Language|GradeLadderName
MERGE|GradeLadderTranslation|VISION|GRD_LDR_OFF|1951/01/01|FR|Officier
```

Deleting Grade Ladders

You can delete a Grade Ladder object using HCM Data Loader, provided that it's not referenced by another object, such as Assignment. Deleting a grade ladder doesn't delete the associated grades. However, if you delete a grade ladder with

steps, then the associated step rates and step-rate values are deleted automatically. This example GradeLadder.dat file deletes an existing grade ladder. It identifies the grade ladder by its source key.

```
METADATA|GradeLadder|SourceSystemOwner|SourceSystemId|EffectiveStartDate
DELETE|GradeLadder|VISION|GRD_LDR_OFF|1951/01/01
```

Related Topics

- Overview of Loading Work Structures

Guidelines for Loading Grade Rates

A grade rate is a collection of general information about a payment type, such as salary, bonus, or overtime. It includes details such as the currency, frequency of the payment, and annualization factor.

For example, the Overtime Rate Analysts grade rate could provide this information for the overtime payments made to analysts. Grade rates are associated with grade-rate values. Each associated grade-rate value identifies a grade and either a fixed payment or a payment range for the grade. For a payment range, you must supply both the minimum and maximum values. This topic describes aspects of the Grade Rate object that you must understand to load grade rates successfully.

Loading Referenced Objects

Grade rates and grade-rate values are created for a legislative data group. Therefore, Legislative Data Group objects must exist before you load grade rates.

Loading Translated Grade-Rate Names

Supply grade-rate names in the language of the user who's loading them. You supply a GradeRateTranslation.dat file to translate grade-rate names into other enabled languages once the grade rates exist. This example translates the name of an existing grade rate. It identifies the grade rate by its source key.

```
METADATA|GradeRateTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Language|GradeRateName
MERGE|GradeRateTranslation|VISION|GR_CIV_SAL|2000/01/01|FR|Salaire civile
```

Deleting Grade Rates

You can delete a Grade Rate object using HCM Data Loader, provided that it's not referenced by another object, such as Salary Basis. If you delete a grade rate, then its associated grade-rate values are deleted automatically. This example GradeRate.dat file deletes an existing grade rate and its grade-rate values. It identifies the grade rate by its source key.

```
METADATA|GradeRate|SourceSystemOwner|SourceSystemId|EffectiveStartDate
DELETE|GradeRate|VISION|GR_CIV_SAL|2000/01/01
```

Related Topics

- Overview of Loading Work Structures

Guidelines for Loading Jobs

A job defines the worker's role in an organization in general terms. For example, a worker's job could be payroll manager, sales consultant, or administrator. This topic describes aspects of the job object that you must understand to load Job objects successfully.

Loading Referenced Objects

If you're associating either job families or valid grades with your jobs, then you must load those objects before you load the jobs.

Benchmark Jobs

To use benchmark jobs:

1. Load any benchmark jobs with the **BenchmarkJobFlag** attribute set to **Yes**. Benchmark jobs can't refer to other benchmark jobs. Therefore, you can't load benchmark jobs with either a **BenchmarkJobId** or **BenchmarkJobCode** value.
2. Load the remaining jobs. For those jobs that refer to a benchmark job, use the **BenchmarkJobId** or **BenchmarkJobCode** attribute to identify the relevant benchmark job. A job can't be its own benchmark job. Therefore, these values must not identify the job that you're loading.

Loading Translated Job Names

Supply job names in the language of the user who's loading them. You supply a JobTranslation.dat file to translate job names into other enabled languages once the jobs exist. This example translates the names of existing jobs. It identifies those jobs by their source keys.

```
METADATA|JobTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Language|Name
MERGE|JobTranslation|VISION|SALES_CONS|2000/01/01|FR|Consultant en Ventes
MERGE|JobTranslation|VISION|SALES_MGR|2000/01/01|FR|Superviseur des Ventes
MERGE|JobTranslation|VISION|SALES_DIR|2000/01/01|FR|Directeur des Ventes
```

Deleting Jobs

You can't delete Job objects using HCM Data Loader. However, you can make jobs inactive. This example Job.dat file inactivates an existing job. It identifies the job by its source key.

```
METADATA|Job|SourceSystemOwner|SourceSystemId|EffectiveStartDate|ActiveStatus
MERGE|Job|VISION|MRKT_CONS|2015/01/01|I
```

Prevent Inflight Job Updates

You can prevent inflight job updates when it's updated using the update and correct flows, or when a date effective department record is deleted, when approvals are pending. You need to create and set the HR_DISABLE_PENDING_APPROVALS_CHECK_IN _HCM_DATA_LOADER profile option to No, if it's not already created to prevent inflight job updates.

Related Topics

- Overview of Loading Work Structures

Examples of Loading Job Families

A job family groups similar or related jobs for ease of reporting. For example, you could add the Trust Analyst and Operations Analyst jobs to the Analyst job family. This topic provides some examples showing how to load and manage Job Family objects.

Creating Job Families Using Source Keys

This example JobFamily.dat file creates several job families that are identified using source keys.

```
METADATA|JobFamily|SourceSystemOwner|SourceSystemId|EffectiveStartDate|JobFamilyCode|JobFamilyName|
ActiveStatus
MERGE|JobFamily|VISION|ANALYST|2000/01/01|ANALYST|Analyst|A
MERGE|JobFamily|VISION|TECHNICIAN|2000/01/01|TECHNICIAN|Technician|A
MERGE|JobFamily|VISION|CONSULT|2000/01/01|CONSULTANT|Consultant|A
MERGE|JobFamily|VISION|MANAGEMENT|2000/01/01|MANAGEMENT|Management|A
```

Loading Translated Job-Family Names

Job family names must be unique in the enterprise. Supply job family names in the language of the user who's loading them. You supply a JobFamilyTranslation.dat file to translate job-family names into other enabled languages once the job families exist. This example translates the names of existing job families. The job families are identified by their source keys.

```
METADATA|JobFamilyTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Language|JobFamilyName
MERGE|JobFamilyTranslation|VISION|ANALYST|2000/01/01|FR|Analyste
MERGE|JobFamilyTranslation|VISION|TECHNICIAN|2000/01/01|FR|Technicien
```

Deleting Job Families

You can delete a Job Family object using HCM Data Loader, provided that the job family isn't referenced by any job. This example deletes a job family. It identifies the job family by its source key.

```
METADATA|JobFamily|SourceSystemOwner|SourceSystemId|EffectiveStartDate
DELETE|JobFamily|VISION|TECHNICIAN|2000/01/01
```

Related Topics

- Overview of Loading Work Structures

Guidelines for Loading Locations

A location is the physical address of a place where you conduct business or that's of interest to your business. The Location object records the address and other details, such as the location name, description, phones, and billing and shipping information.

This topic describes aspects of the Location object that you must understand to load locations successfully.

Changing Logical Start and End Dates for Locations

You can't change either the first effective start date or last effective end date for an existing location. Create location records with effective start dates on or before the start dates of other objects, such as Organization, that refer to your locations.

Loading Translated Location Names

Supply location names and descriptions in the language of the user who's loading them. You supply a LocationTranslation.dat file to translate location names and descriptions into other enabled languages once the locations exist. This example translates the name and description of an existing location. It identifies the location by its source key.

```
METADATA|LocationTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Language|LocationName|
Description
MERGE|LocationTranslation|VISION|LOC_RD_UK|1951/01/01|FR|Vision Corporation Recherche et Developpement|
Centre de recherche et de developpement
```

Deleting Locations

You can't delete Location objects using HCM Data Loader. However, you can make locations inactive. This example Location.dat file makes a location inactive as of 01/01/15. The location is identified by its source key.

```
METADATA|Location|SourceSystemOwner|SourceSystemId|EffectiveStartDate|ActiveStatus
MERGE|Location|VISION|LOC_UK_100|2015/01/01|I
```

Loading Legal Employer- Location Association

You can associate legal employers with a location using the **Legal Employers Operating At This Location** extensible flexfield (EFF) context. First, you need to add this context and then associate the legal employers with the location. This example loads the EFF for a location.

```
METADATA|Location|FLEX:PER_LOCATIONS_DF|LocationId|SetId|SetCode|LocationImageURL|ActiveStatus|
MainphoneAreaCode|MainphoneCountryCode|MainphoneExtension|MainphoneSubscriberNumber|FaxAreaCode|
FaxCountryCode|FaxExtension|FaxSubscriberNumber|OtherphoneAreaCode|OtherphoneCountryCode|
OtherphoneExtension|OtherphoneSubscriberNumber|OfficialLanguageCode|EmailAddress|ShipToSiteFlag|
ShipToLocationSetCode|ShipToLocationCode|ShipToLocationId|ReceivingSiteFlag|BillToSiteFlag|OfficeSiteFlag|
DesignatedReceiverId|DesignatedPersonNumber|InventoryOrganizationId|InventoryOrganizationName|
GeoHierarchyNodeId|GeoHierarchyNodeCode|LocationCode|LocationName|Description|AddressLine1|
AddressLine2|AddressLine3|AddressLine4|Building|Country|FloorNumber|LongPostalCode|PostalCode|
Region1|Region2|Region3|TimezoneCode|TownOrCity|AddlAddressAttribute1|AddlAddressAttribute2|
AddlAddressAttribute3|AddlAddressAttribute4|AddlAddressAttribute5|EffectiveStartDate|EffectiveEndDate|
CategoryCode|ActionReasonCode|SourceSystemOwner|SourceSystemId|GUID|SourceRefTableName=(source-
table-name)|SourceRef001=(source-column-001)|SourceRef002=(source-column-002)|SourceRef003=(source-
column-003)|SourceRef004=(source-column-004)|SourceRef005=(source-column-005)|SourceRef006=(source-
column-006)|SourceRef007=(source-column-007)|SourceRef008=(source-column-008)|SourceRef009=(source-
column-009)|SourceRef010=(source-column-010)|locglobal(PER_LOCATIONS_DF=Global Data Elements)|
locationcntx(PER_LOCATIONS_DF=location_extra info)
METADATA|LocationOtherAddress|LocAddressUsageId|EffectiveStartDate|EffectiveEndDate|AddressUsageType|
LocationId(SourceSystemId)|LocationSetCode|LocationCode|AddressLine1|AddressLine2|AddressLine3|
AddressLine4|Building|TownOrCity|FloorNumber|Country|PostalCode|LongPostalCode|Region1|Region2|
Region3|TimezoneCode|SourceSystemOwner|SourceSystemId|GUID|SourceRefTableName=(source-table-
name)|SourceRef001=(source-column-001)|SourceRef002=(source-column-002)|SourceRef003=(source-
column-003)|SourceRef004=(source-column-004)|SourceRef005=(source-column-005)|SourceRef006=(source-
column-006)|SourceRef007=(source-column-007)|SourceRef008=(source-column-008)|SourceRef009=(source-
column-009)|SourceRef010=(source-column-010)METADATA|LocationLegislative|FLEX:PER_LOCATION_LEG_EFF|
EFF_CATEGORY_CODE|LocationLegId|EffectiveStartDate|EffectiveEndDate|LocationId(SourceSystemId)|
ActionOccurrenceId|LegislationCode|LleInformationCategory|SequenceNumber|SourceSystemOwner|
SourceSystemId|GUID|SetId|SetCode|LocationCode|locationsegment1(PER_LOCATION_LEG_EFF=Location_Context)|
locationsegment2(PER_LOCATION_LEG_EFF=Location_Context)
```

```
METADATA|LocationExtraInfo|FLEX:PER_LOCATION_INFORMATION_EFF|EFF_CATEGORY_CODE|LocationExtraInfoId|
EffectiveStartDate|EffectiveEndDate|LocationId(SourceSystemId)|InformationType|ActionOccurrenceId|
LeiInformationCategory|SequenceNumber|LegislationCode|SourceSystemOwner|SourceSystemId|GUID|SetId|SetCode|
LocationCode|testseg(PER_LOCATION_INFORMATION_EFF=Test_HDL)
```

This example associates legal employers for a specific location context.

```
MERGE|Location||||COMMON||A|||||||||||||||||||||||||||Location_{{xs}}_021|Location_{{xs}}_021|Description|
ABC|AddressLine2|AddressLine3|AddressLine4|Building|US|FloorNumber||10001|New York|NY|||Manhattan||||||
2000/01/01|2004/12/31|||HCMQA-001|Location_{{xs}}_021|||||||||||||
MERGE|Location||||COMMON||I|||||||||||||||||||||||||||Location_{{xs}}_021|Location_{{xs}}_021|Description|
DEF|AddressLine2|AddressLine3|AddressLine4|Building|US|FloorNumber||10001|New York|NY|||Manhattan||||||
2005/01/01|2009/12/31|||HCMQA-001|Location_{{xs}}_021|||||||||||||
MERGE|Location||||COMMON||A|||||||||||||||||||||||||||Location_{{xs}}_021|Location_{{xs}}_021|Description|
DEF|AddressLine2|AddressLine3|AddressLine4|Building|US|FloorNumber||10001|New York|NY|||Manhattan||||||
2010/01/01|4712/12/31|||HCMQA-001|Location_{{xs}}_021|||||||||||||
MERGE|LocationOtherAddress||2000/01/01|2004/12/31|MAILING|Location_{{xs}}_021|||AddressLine1|AddressLine2|
AddressLine3|AddressLine4|Building|Atlantic|1st floor|US|16111||Crawford|PA|Region3||HCMQA-001|
LocationAddr_{{xs}}_021|||||||||||||
MERGE|LocationOtherAddress||2005/01/01|2009/12/31|MAILING|Location_{{xs}}_021|||AddressLine1Updated|
AddressLine2 Updated|Updated AddressLine3|AddressLine4 Updated|Building|Atlantic|1st floor|US|16111||
Crawford|PA|Region3||HCMQA-001|LocationAddr_{{xs}}_021|||||||||||||
MERGE|LocationOtherAddress||2010/01/01|4712/12/31|MAILING|Location_{{xs}}_021|||AddressLine1|AddressLine2|
AddressLine3|AddressLine4|Building|Atlantic|1st floor|US|16111||Crawford|PA|Region3||HCMQA-001|
LocationAddr_{{xs}}_021|||||||||||||
MERGE|LocationLegislative|Location_Context|HCM_LOC_LEG||2000/01/01|2004/12/31|Location_{{xs}}_021||US|||
HCMQA-001|LocationLeg_{{xs}}_021|||COMMON|Location_{{xs}}_021|abc|123
MERGE|LocationLegislative|Location_Context|HCM_LOC_LEG||2005/01/01|2009/12/31|Location_{{xs}}_021||US|||
HCMQA-001|LocationLeg_{{xs}}_021|||COMMON|Location_{{xs}}_021|def|456
MERGE|LocationLegislative|Location_Context|HCM_LOC_LEG||2010/01/01|4712/12/31|Location_{{xs}}_021||US|||
HCMQA-001|LocationLeg_{{xs}}_021|||COMMON|Location_{{xs}}_021|ghi|789
MERGE|LocationExtraInfo|Test_HDL|HCM_LOC||2000/01/01|2004/12/31|Location_{{xs}}_021|Test_HDL||Test_HDL||US|
HCMQA-001|LocationEIT_{{xs}}_021|||COMMON||abc
MERGE|LocationExtraInfo|Test_HDL|HCM_LOC||2005/01/01|2009/12/31|Location_{{xs}}_021|Test_HDL||Test_HDL||US|
HCMQA-001|LocationEIT_{{xs}}_021|||COMMON||
MERGE|LocationExtraInfo|Test_HDL|HCM_LOC||2010/01/01|4712/12/31|Location_{{xs}}_021|Test_HDL||Test_HDL||US|
HCMQA-001|LocationEIT_{{xs}}_021|||COMMON||xyz
```

Prevent Inflight Location Updates

You can prevent inflight location updates when it's updated using the update and correct flows, or when a date effective location record is deleted, when approvals are pending. You need to create and set the HR_DISABLE_PENDING_APPROVALS_CHECK_IN _HCM_DATA_LOADER profile option to No, if it's not already created to prevent inflight location updates.

Related Topics

- Overview of Loading Work Structures

Guidelines for Loading Organizations

An organization can be any organizational unit, such as a department or division, in your enterprise. Organizations provide the framework for performing legal reporting, financial control, and management reporting.

This topic describes aspects of the Organization object that you must understand to load organizations successfully.

Organization Components

A single organization can have multiple classifications. For example, a division can also be a department. Therefore, you can load multiple Organization Classification components for a single Organization object.

> **Note:** You must load any Organization Extra Information component with the parent Organization component in the same Organization.dat file. You can't load Organization Extra Information components independently of the parent Organization component. You can load other child components either when creating the organization or once the organization exists.

Loading Referenced Objects

Before you load organizations, referenced objects must exist in the target environment. In particular:

- If you plan to associate locations with your organizations, then you must load the locations before you load the organizations.
- To associate a reporting manager with your organization, you must:
  a. Load the organization without identifying the reporting manager.
  b. Load the worker information for the reporting manager.
  c. Update your organization to add the reporting manager.

  > **Tip:** As a worker is hired into a legal employer, business unit, and department, you can't load workers before you load organizations.

> **Note:** Departments are set enabled. You must ensure that relevant reference data sets exist before you load departments. Other types of organizations aren't set enabled.

Organization Classification Codes

An organization can have one or more classifications. You can supply either the code or its name. Valid classifications are shown in this table.

Classification Name	Classification Code
Department	DEPARTMENT
Disability Organization	HCM_DISABILITY_ORGANIZATION
Division	HCM_DIVISION
Reporting Establishment	HCM_REPORTING_ESTABLISHMENT
Union	PER_WORKER_UNION

You load an Organization Classification component for each of the organization's classifications, including the one identified on the Organization component.

Creating Departments Using Source Keys

This example Organization.dat file creates a department that is identified by source keys.

```
METADATA|Organization|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Name|ClassificationCode
MERGE|Organization|VISION|SALES|1980/01/01|Sales|DEPARTMENT
METADATA|OrgUnitClassification|SourceSystemOwner|SourceSystemId|OrganizationId(SourceSystemId)|
EffectiveStartDate|ClassificationCode|SetCode|Status
MERGE|OrgUnitClassification|VISION|SALES_DEP|SALES|1980/01/01|DEPARTMENT|COMMON|A
```

Loading Translated Organization Names

Supply organization names in the language of the user who's loading them. You supply an OrganizationTranslation.dat file to translate organization names into other enabled languages once the organizations exist. This example OrganizationTranslation.dat file translates the name of an existing organization. It identifies the organization by its source key.

```
METADATA|OrganizationTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Language|Name
MERGE|OrganizationTranslation|VISION|MRKT|2000/01/01|FR|Marketing
```

Changing the Logical Start of Organizations

You can't change the end date for an organization, instead make it inactive. However, you can change the earliest start date for an organization. This example Organization.dat file adjusts the earliest start date of a department that is identified by source keys.

```
METADATA|Organization|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Name|ClassificationCode|
ReplaceFirstEffectiveStartDate
MERGE|Organization|VISION|SALES|1950/01/01|Sales|DEPARTMENT|Y
METADATA|OrgUnitClassification|SourceSystemOwner|SourceSystemId|OrganizationId(SourceSystemId)|
EffectiveStartDate|ClassificationCode|SetCode|Status|ReplaceFirstEffectiveStartDate
MERGE|OrgUnitClassification|VISION|SALES_DEP|SALES|1950/01/01|DEPARTMENT|COMMON|A|Y
```

Note: You must supply the classification record. Your organization can't pre-date its classification.

Deleting Organizations

You can't delete Organization objects using HCM Data Loader, nor can you make them inactive. However, you can make an organization classification inactive, provided that it's no longer used. For example, for an organization that's classified as both a department and a division, you could:

- Make the division inactive but leave the department active, so that you can still continue to use the department.
- Deactivate the whole organization by making both classifications inactive.

This example Organization.dat file inactivates an existing organization classification. The organization is identified by its source key.

```
METADATA|Organization|SourceSystemOwner|SourceSystemId|EffectiveStartDate
MERGE|Organization|VISION|CLARITY|2015/01/01
METADATA|OrgUnitClassification|SourceSystemOwner|SourceSystemId|EffectiveStartDate|
OrganizationId(SourceSystemId)|Status
MERGE|OrgUnitClassification|VISION|CLARITY_DIS|2015/01/01|CLARITY|I
```

Note: The file must include an Organization component with an effective start date equal to the date on which the classification becomes inactive.

Prevent Inflight Department Updates

You can prevent inflight department updates when it's updated using the update and correct flows, or when a date effective department record is deleted, when approvals are pending. You need to create and set the HR_DISABLE_PENDING_APPROVALS_CHECK_IN _HCM_DATA_LOADER profile option to No, if it's not already created to prevent inflight department updates.

Related Topics

- Overview of Loading Work Structures

Guidelines for Loading Positions

A position is a single occurrence of a job in a specified department. Positions may also be restricted by location. For example, you could create the HR Manager Sales France position for the HR Manager job in the Sales department in France.

This topic describes aspects of the Position object that you must understand to load positions successfully.

Effective Start Dates

A Position object and its child components must have the same effective start date.

Loading Referenced Objects

Before you load positions, referenced objects must exist in the target environment. In particular:

- Positions exist in business units. Ensure that referenced business units exist.
- Both **Job** and **Department** are required attributes of a position. Ensure that referenced jobs and departments exist.
- You can associate a location and valid grades with a position. Ensure that any referenced locations and grades exist.

Position Type

When the position type is **Single incumbent**, you must supply:

- An FTE value. The value must be less than or equal to 1. For example, if the position incumbent works half the standard working hours, then the full-time equivalent value is 0.5.
- A headcount of 1.

When the position type is **Shared**, you must supply:

- An FTE value
- A headcount that's equal to or greater than 1

Note: When loading positions, the position code can be automatically generated, hence we recommend you leave the position code blank in the dat file to avoid duplicate position codes being generated.

Loading Translated Position Names

Supply position names in the language of the user who's loading them. You supply a PositionTranslation.dat file to translate position names into other enabled languages once the positions exist. This example PositionTranslation.dat file translates the name of an existing position. It identifies the position by its source key.

```
METADATA|PositionTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Language|Name
MERGE|PositionTranslation|VISION|APP_SALES|2015/01/01|FR|Sales Consultant d'application
```

Deleting Positions

You can delete a Position object using HCM Data Loader, provided that it has never had an incumbent. When you delete a position, its child components are deleted automatically. This example Position.dat file deletes an existing position. The position is identified by its source key.

```
METADATA|Position|SourceSystemOwner|SourceSystemId|EffectiveStartDate
DELETE|Position|VISION|APP_SALES|2015/01/01
```

Prevent Inflight Position Updates

You can prevent inflight position updates when it's updated using the Request a Position Change, update, or correct flows, or when a date effective position record is deleted, when approvals are pending. You need to create and set the HR_DISABLE_PENDING_APPROVALS_CHECK_IN _HCM_DATA_LOADER profile option to **No**, if it's not already created; to prevent inflight position updates.

Related Topics

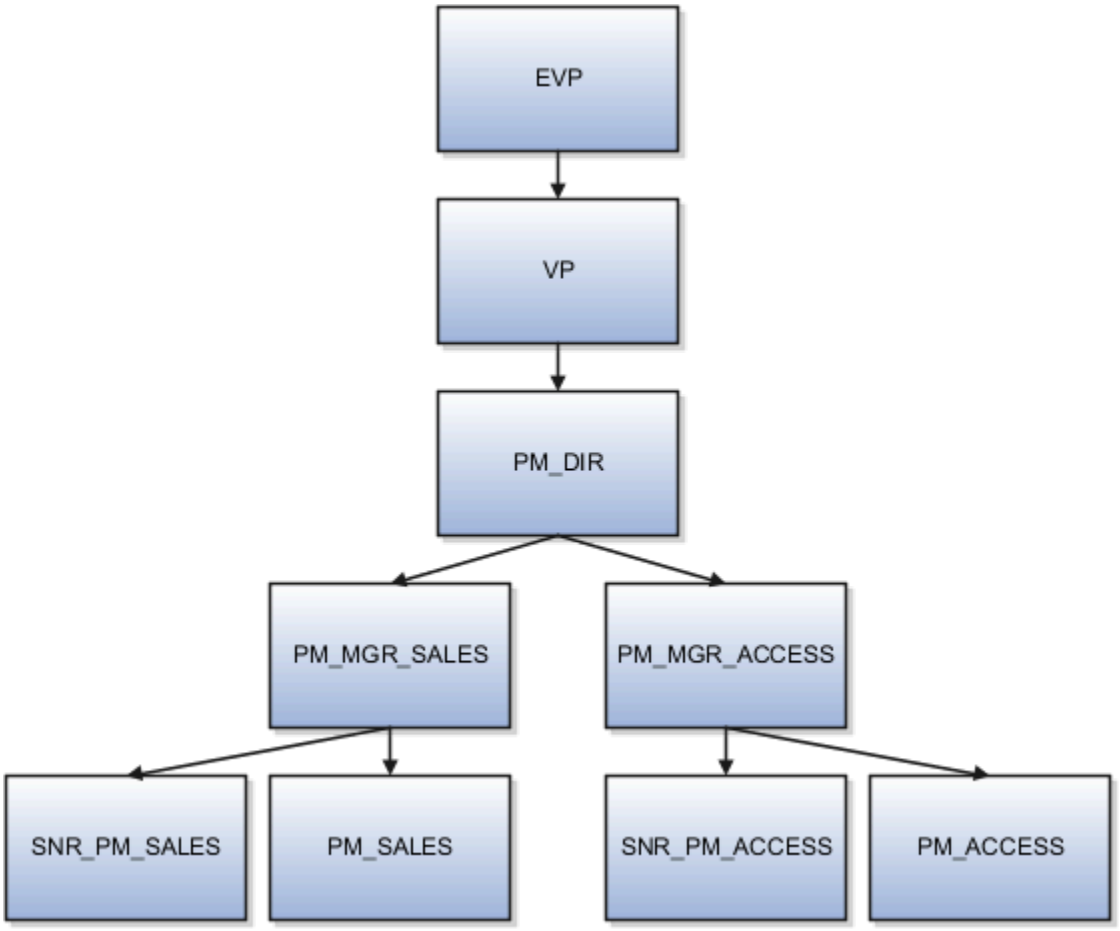
- Overview of Loading Work Structures

Examples of Loading HCM Position Hierarchies

An HCM position hierarchy is derived from the relationships between positions and their parent positions. You can identify a position's parent position on the Manage Positions page if the Use **HCM Position Hierarchy** option is enabled for the enterprise.

This topic provides examples of how to load Position Hierarchy objects using HCM Data Loader. Each Position Hierarchy object identifies a position and its parent. All positions must exist in the target environment.

This figure shows the position hierarchy that these examples load.



Loading HCM Position Hierarchies Using Source Keys

This example PositionHierarchy.dat file loads Position Hierarchy objects using source keys.

```
METADATA|PositionHierarchy|SourceSystemOwner|SourceSystemId|EffectiveStartDate|EffectiveEndDate|
ParentPositionId(SourceSystemId)|PositionId(SourceSystemId)
MERGE|PositionHierarchy|Vision|1|2015/01/01|4712/12/31|PM_MGR_ACCESS|PM_ACCESS
MERGE|PositionHierarchy|Vision|2|2015/01/01|4712/12/31|PM_MGR_ACCESS|SNR_PM_ACCESS
MERGE|PositionHierarchy|Vision|3|2015/01/01|4712/12/31|PM_MGR_SALES|PM_SALES
MERGE|PositionHierarchy|Vision|4|2015/01/01|4712/12/31|PM_MGR_SALES|SNR_PM_SALES
MERGE|PositionHierarchy|Vision|5|2015/01/01|4712/12/31|PM_DIR|PM_MGR_ACCESS
MERGE|PositionHierarchy|Vision|6|2015/01/01|4712/12/31|PM_DIR|PM_MGR_SALES
MERGE|PositionHierarchy|Vision|7|2015/01/01|4712/12/31|VP|PM_DIR
MERGE|PositionHierarchy|Vision|8|2015/01/01|4712/12/31|EVP|VP
```

Loading HCM Position Hierarchies Using User Keys

This example PositionHierarchy.dat file loads Position Hierarchy objects using user keys.

```
METADATA|PositionHierarchy|EffectiveStartDate|EffectiveEndDate|ParentBusinessUnitName|ParentPositionCode|
BusinessUnitName|PositionCode
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_MGR_ACCESS|VisionBU|PM_ACCESS
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_MGR_ACCESS|VisionBU|SNR_PM_ACCESS
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_MGR_SALES|VisionBU|PM_SALES
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_MGR_SALES|VisionBU|SNR_PM_SALES
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_DIR|VisionBU|PM_MGR_ACCESS
```

```
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_DIR|VisionBU|PM_MGR_SALES
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|VP|VisionBU|PM_DIR
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|EVP|VisionBU|VP
```

Running the Flatten HCM Position Hierarchy Process

Changes to the parent position on the Manage Positions page automatically trigger a process to update the position hierarchy. This process is also triggered automatically when you load position hierarchies using HCM Data Loader. You can prevent the **Flatten HCM Position Hierarchy** process from running automatically by including a SET instruction in the PositionHierarchy.dat file.

Examples of Loading Position Budgets

Position Budgeting helps to capture the budgetary values and to enforce them while creating or updating budgets for positions. You can now define a budget period and measure your activity against these values.

A position budget table, budget configuration options, and a mechanism to load and store position budgets using HDL is provided. You can use this budget information to alert the users when they exceed the position budget. You can load budget definitions by using the PositionBudget.dat file in HDL. Once the budget data is loaded in the application, the validations for FTE, headcount, and budget amount take effect when a position is created or updated.

This topic provides examples of how to create, update, and delete position budgets using HCM Data Loader.

Creating Position Budgets

You can create position budgets using HCM Data Loader. This example PositionBudget.dat file creates a position budget.

```
METADATA|PositionBudget|Name|Code|BusinessUnitId|BusinessUnitName|DepartmentId|DepartmentName|LocationId|
LocationSetCode|LocationCode|AllocatedFTE|AllocatedHeadcount|AllocatedAmount|BudgetPeriodStartYear|
SourceSystemOwner|SourceSystemId|GUID
MERGE|PositionBudget|CARDIO BUDGET|VISCARDBUDGET||Vision Corporation Enterprise||Cardiology||COMMON|
VISION_UNIVERSITY_0_2451976074357|200|200|100000|2023|HCMQA-001|MB_0090|
```

Updating Position Budgets

You can update position budgets using HCM Data Loader. This example PositionBudget.dat file updates a position budget.

```
METADATA|PositionBudget|Name|Code|BusinessUnitId|BusinessUnitName|DepartmentId|DepartmentName|LocationId|
LocationSetCode|LocationCode|AllocatedFTE|AllocatedHeadcount|AllocatedAmount|BudgetPeriodStartYear|
SourceSystemOwner|SourceSystemId|GUID
MERGE|PositionBudget|CARDIO BUDGET|VISCARDBUDGET||||||||300|250|120000|2023|HCMQA-001|MB_0090|
```

Deleting Position Budgets

You can delete position budgets using HCM Data Loader. This example PositionBudget.dat file deletes a position budget.

```
METADATA|PositionBudget|Name|Code|BusinessUnitId|BusinessUnitName|DepartmentId|DepartmentName|LocationId|
LocationSetCode|LocationCode|AllocatedFTE|AllocatedHeadcount|AllocatedAmount|BudgetPeriodStartYear|
SourceSystemOwner|SourceSystemId|GUID
DELETE|PositionBudget|CARDIO BUDGET|VISCARDBUDGET||||||||||2023|HCMQA-001|MB_0090|
```

Guidelines for Loading Department Trees

A department tree is a hierarchy of departments. For each tree, you can define multiple versions. However, only one version can be active on any date.

This topic describes aspects of Department Tree objects that you must understand to load them successfully using HCM Data Loader.

How Department Trees Are Implemented

Department trees are implemented using two HCM Data Loader objects. This table introduces those objects.

Object	Description
Department Tree	The definition of a tree and the versions of the tree. A tree version is empty until you load department tree nodes for the tree version.
Department Tree Node	The department tree nodes for a tree version. Each node represents a department.

Preparing Department Trees for Use

After loading department tree nodes, you must flatten, audit, and activate your tree version. You can't reference your tree version from other objects, such as Area of Responsibility, until it's active. You can activate your tree version on the Manage Department Trees page in the Workforce Structures work area. In summary, you:

1. Search for and select your department tree.
2. Expand the hierarchy and select the version to activate.
3. Select **Actions** > **Set Status** > **Active** to activate the tree version. The tree flattening and audit occur automatically.

For more information about managing hierarchy trees, see the Implementing Global Human Resources guide.

Loading Department Trees

You supply department-tree data in the DepartmentTree.dat file for processing by HCM Data Loader. The **TreeStructureCode** attribute of both the department tree and the department tree version must be set to **PER_DEPT_TREE_STRUCTURE** for a department tree.

> **Note:** When you load a new department tree version for an existing department tree, you must include the Department Tree component in the DepartmentTree.dat file. You can't load a new department tree version in isolation.

```
METADATA|DepartmentTreeNode|TreeStructureCode|TreeCode|TreeVersionName|DepartmentId(SourceSystemOwner)|
DepartmentId(SourceSystemId)  |DepartmentName|ParentDepartmentId(SourceSystemOwner)|
ParentDepartmentId(SourceSystemId)|ParentDepartmentName  |SourceSystemOwner|SourceSystemIdMERGE|
DepartmentTreeNode|PER_DEPT_TREE_STRUCTURE|AdministrativeHierarchy|AdministrativeHierarchy_December2020  |
HRC_SQLLOADER|AdministrativeHierarchy_Org_SS_ID_000001||||||
```

```
MERGE|DepartmentTreeNode|PER_DEPT_TREE_STRUCTURE|AdministrativeHierarchy |
AdministrativeHierarchy_December2020|HRC_SQLLOADER|AdministrativeHierarchy_Org_SS_ID_000002||HRC_SQLLOADER|
AdministrativeHierarchy_Org_SS_ID_000001|||
```

Deleting Department Trees

You can delete Department Tree and Department Tree Version components using HCM Data Loader. When you delete a department tree, the tree and all of its versions and nodes are deleted.

Note: The departments themselves aren't deleted. Only the relationships defined in the tree structure are deleted.

This example DepartmentTree.dat file deletes a department tree:

```
METADATA|DepartmentTree|TreeStructureCode|TreeCode
DELETE|DepartmentTree|PER_DEPT_TREE_STRUCTURE|Department Tree Code
```

If you delete a department tree version, then only the specified version and its nodes are deleted. The tree and its other versions are retained. This example DepartmentTree.dat file deletes a department tree version:

```
METADATA|DepartmentTreeVersion|TreeStructureCode|TreeCode|TreeVersionName
DELETE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Feb
```

Related Topics

- Guidelines for Loading Department Tree Nodes

Guidelines for Loading Department Tree Nodes

A department is an internal organization to which workers are assigned. You can add any organization that has the Department classification as a tree node to a department tree version.

This topic describes aspects of Department Tree Node objects that you must understand to load them successfully using HCM Data Loader.

Preparing to Load Department Tree Nodes

Before you load department tree nodes:

- You must create the department tree and department tree version to which the nodes belong. You can either load them using HCM Data Loader or create them using the **Manage Department Trees** task in the Workforce Structures work area.
- The departments that the tree nodes refer to must exist on the start date of the tree hierarchy version. Simply adding a department to the department tree hierarchy doesn't create the department. Similarly, removing a department reference from a department tree hierarchy doesn't delete the department record.

Loading Department Tree Nodes

The department tree node records the relationship between a parent department and a child, which can be either a department or another department tree version. When the child is another department tree version, the department nodes of that version are included by reference. When you load department tree nodes, these rules apply:

- You must supply a parent department for all nodes except the top-level node of your department tree version. The parent departments must exist.
- The **TreeStructureCode** attribute must be set to **PER_DEPT_TREE_STRUCTURE**.
- To avoid circular relationships, a department can appear only once in each tree version.
- Each tree version can have only one top node. If you need multiple peer departments at the top, then consider using a dummy department as the top node.

Note: When you load tree nodes, nodes for the same tree and version will be grouped, ordered and processed single-threaded. This is to ensure that the parent node is created before the child nodes that reference it. When you load a large number of nodes for the same tree and version, the load time will be significantly increased, due to single-threaded processing.

Related Topics

- Examples of Loading Department Tree Nodes
- Guidelines for Loading Department Trees
- Processes to Run After Loading Data

Examples of Loading Department Tree Nodes

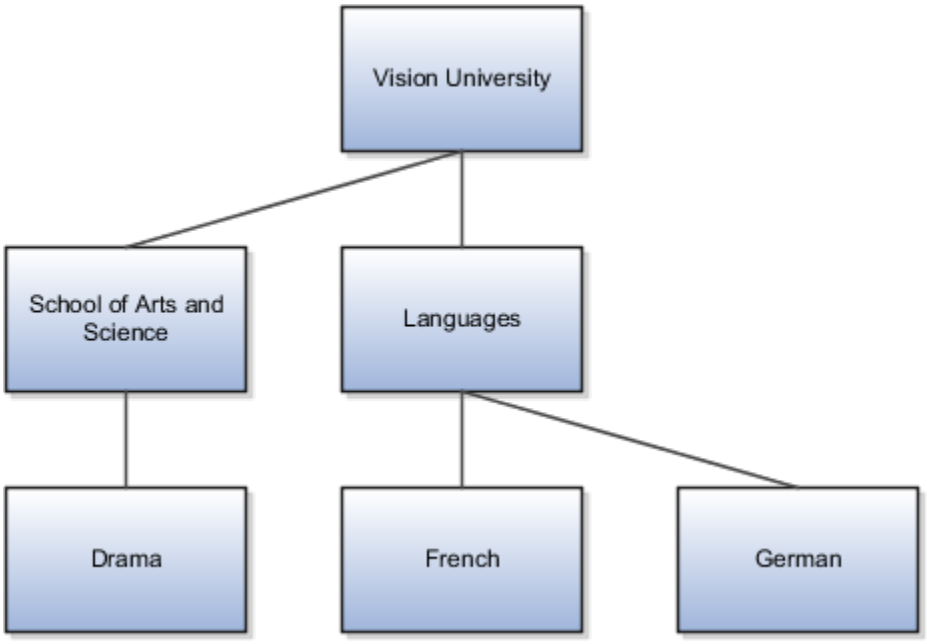
You can load any organization that has the Department classification as a tree node to a department tree version. This topic shows how to load Department Tree Node objects using HCM Data Loader.

Creating Department Tree Nodes

This example DepartmentTreeNode.dat file loads department tree nodes to an existing department tree version.

```
METADATA|DepartmentTreeNode|TreeStructureCode|TreeCode|TreeVersionName|DepartmentName|ParentDepartmentName
MERGE|DepartmentTreeNode|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|Vision University|
MERGE|DepartmentTreeNode|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|School of Arts and
 Science|Vision University
MERGE|DepartmentTreeNode|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|Drama|School of Arts
 and Science
MERGE|DepartmentTreeNode|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|Languages|Vision
 University
MERGE|DepartmentTreeNode|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|French|Languages
MERGE|DepartmentTreeNode|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|German|Languages
```

This figure shows the hierarchy nodes that this DepartmentTreeNode.dat file creates.

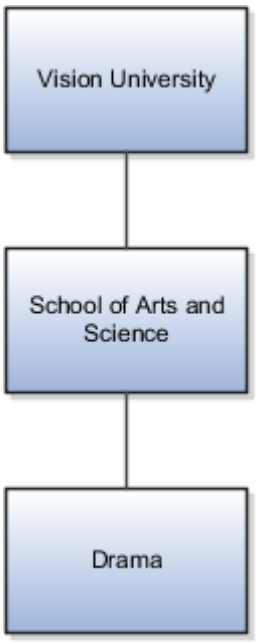


Deleting Department Tree Nodes

You can delete Department Tree Node objects using HCM Data Loader. The departments are removed from the tree version, but the department objects themselves continue to exist. To delete a department tree node and all nodes below it in the department tree hierarchy, set the **DeleteChildNodesFlag** attribute to **Y**. This example DepartmentTreeNode.dat file deletes a node and its child nodes.

```
METADATA|DepartmentTreeNode|TreeStructureCode|TreeCode|TreeVersionName|DepartmentName|DeleteChildNodesFlag
DELETE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|Languages|Y
```

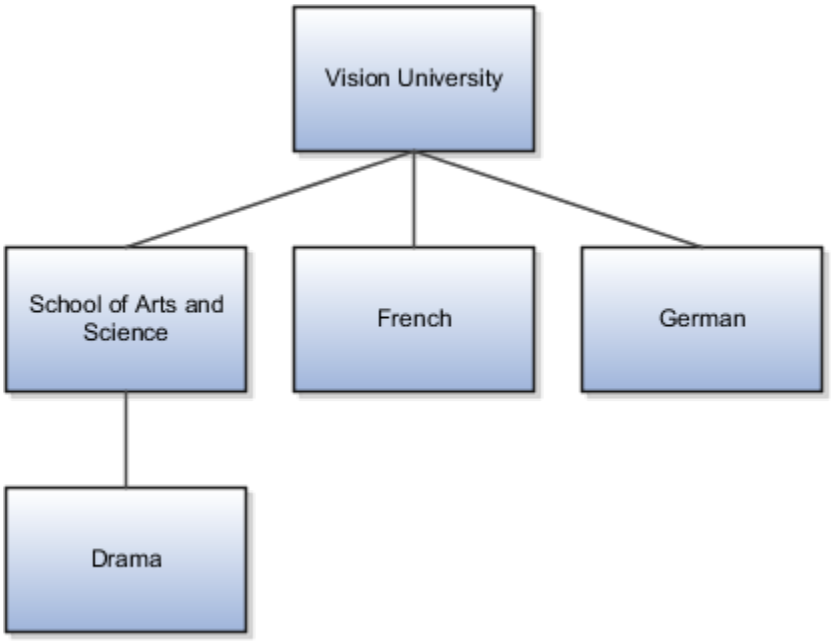
This figure shows the hierarchy after deleting the Languages node and its child nodes.



To delete a department tree node and promote its child nodes, set the **DeleteChildNodesFlag** attribute to **N**. This example DepartmentTreeNode.dat file deletes a node and promotes its child nodes.

```
METADATA|DepartmentTreeNode|TreeStructureCode|TreeCode|TreeVersionName|DepartmentName|DeleteChildNodesFlag
DELETE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|Languages|N
```

This figure shows the hierarchy after deleting the Languages node and promoting its child nodes.



Related Topics

- Guidelines for Loading Department Tree Nodes

Guidelines for Loading Organization Trees

An organization tree is a hierarchy of organizations of any classification. For each tree, you can define multiple versions. However, only one version can be active on any date.

This topic describes aspects of Organization Tree objects that you must understand to load them successfully using HCM Data Loader.

How Organization Trees Are Implemented

Organization trees are implemented using two HCM Data Loader objects. This table introduces those objects.

Object	Description
Organization Tree	The definition of a tree and the versions of the tree. A tree version is empty until you load organization tree nodes for the tree version.
Organization Tree Node	The organization tree nodes for a tree version. Each node represents an organization.

Object	Description

Preparing Organization Trees for Use

After loading organization tree nodes, you must flatten, audit, and activate your tree version. You can't reference your tree version from other objects, such as Area of Responsibility, until it's active. You can activate your tree version on the Manage Organization Trees page in the Workforce Structures work area. In summary, you:

1. Search for and select your organization tree.
2. Expand the hierarchy and select the version to activate.
3. Select **Actions** > **Set Status** > **Active** to activate the tree version. The tree flattening and audit occur automatically.

For more information about managing hierarchy trees, see the Implementing Global Human Resources guide.

Loading Organization Trees

You supply organization-tree data in the OrganizationTree.dat file for processing by HCM Data Loader. The **TreeStructureCode** attribute of both the organization tree and the organization tree version must be set to **PER_ORG_TREE_STRUCTURE** for an organization tree.

> **Note:** When you load a new organization tree version for an existing organization tree, you must include the organization tree component in the OrganizationTree.dat file. You can't load a new organization tree version in isolation.

```
METADATA|OrganizationTreeNode|OrganizationTreeNodeId|TreeStructureCode|TreeCode|TreeVersionName|
OrganizationId(SourceSystemOwner)|OrganizationId(GUID) |OrganizationName |ClassificationCode|
ReferenceTreeVersionName|ReferenceTreeCode|ParentOrganizationId(SourceSystemOwner) |
ParentOrganizationId(GUID)|ParentOrganizationName|ParentClassificationCode|SourceSystemOwner|
SourceSystemIdMERGE |OrganizationTreeNode||PER_ORG_TREE_STRUCTURE |ApprovalHierarchy|
ApprovalHierarchy_January2021|HRC_SQLLOADER|B104F5BF20E83097E053334CF70A512F ||DEPARTMENT||||||||
MERGE|OrganizationTreeNode||PER_ORG_TREE_STRUCTURE|ApprovalHierarchy |ApprovalHierarchy_January2021
 |HRC_SQLLOADER|B14F66EC9CC8DF14E053334CF70A71FF||DEPARTMENT|||HRC_SQLLOADER|
B104F5BF20E83097E053334CF70A512F||DEPARTMENT||
```

Deleting Organization Trees

You can delete Organization Tree and Organization Tree Version components using HCM Data Loader. When you delete an organization tree, the tree and all of its versions and nodes are deleted.

> **Note:** The organizations themselves aren't deleted. Only the relationships defined in the tree structure are deleted.

This example OrganizationTree.dat file deletes an organization tree:

```
METADATA|OrganizationTree|TreeStructureCode|TreeCode
DELETE|OrganizationTree|PER_ORG_TREE_STRUCTURE|Organization Tree Code
```

If you delete an organization tree version, then only the specified version and its nodes are deleted. The tree and its other versions are retained. This example OrganizationTree.dat file deletes an organization tree version:

```
METADATA|OrganizationTreeVersion|TreeStructureCode|TreeCode|TreeVersionName
DELETE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Organization Tree Code|Tree Version Feb
```

Related Topics

- Guidelines for Loading Organization Tree Nodes

Guidelines for Loading Organization Tree Nodes

You can load an organization of any classification as a tree node to an organization tree version. This topic describes aspects of Organization Tree Node objects that you must understand to load them successfully using HCM Data Loader.

Preparing to Load Organization Tree Nodes

Before you load organization tree nodes:

- You must create the organization tree and organization tree version to which the nodes belong. You can either load them using HCM Data Loader or create them using the **Manage Organization Trees** task in the Workforce Structures work area.
- The organizations that the tree nodes refer to must exist on the start date of the tree hierarchy version. Simply adding an organization to the organization tree hierarchy doesn't create the organization. Similarly, removing an organization reference from an organization tree hierarchy doesn't delete the organization record.

Loading Organization Tree Nodes

The organization tree node records the relationship between a parent organization and a child, which can be either an organization or another organization tree version. When the child is an organization, you specify its classification on the **ClassificationCode** attribute. When the child is another organization tree version, the organization nodes of that version are included by reference. When you load organization tree nodes, these rules apply:

- You must supply a parent organization for all nodes except the top-level node of your organization tree version. The parent organizations must exist.
- The **TreeStructureCode** attribute must be set to **PER_ORG_TREE_STRUCTURE**.
- To avoid circular relationships, an organization can appear only once in each tree version.
- Each tree version can have only one top node. If you need multiple peer organizations at the top, then consider using the enterprise as the top node.

> **Note:** When you load tree nodes, nodes for the same tree and version will be grouped, ordered and processed single-threaded. This is to ensure that the parent node is created before the child nodes that reference it. When you load a large number of nodes for the same tree and version, the load time will be significantly increased, due to single-threaded processing.

Related Topics

- Examples of Loading Organization Tree Nodes
- Guidelines for Loading Organization Trees
- Processes to Run After Loading Data

Examples of Loading Organization Tree Nodes

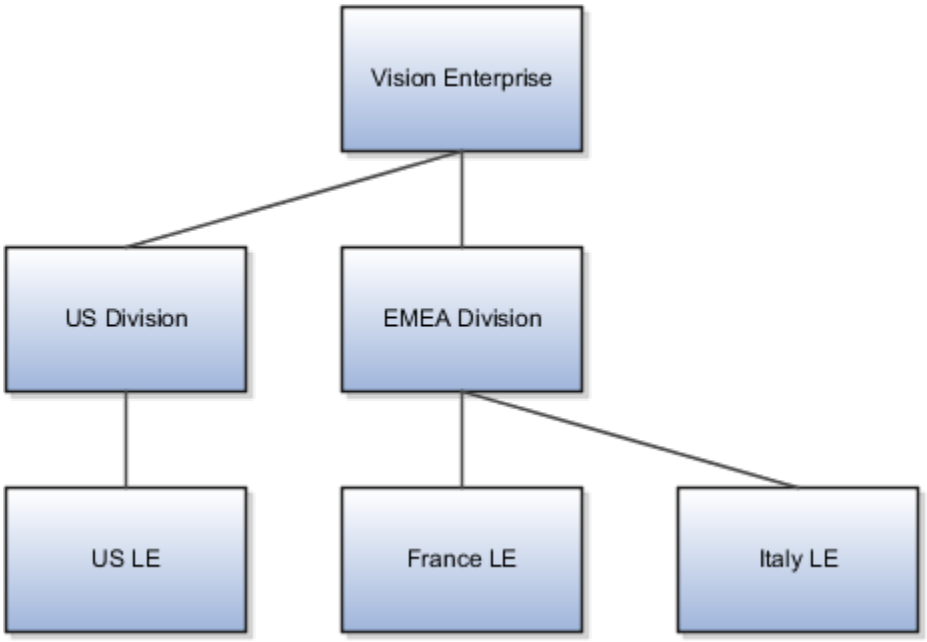
You can load an organization with any classification as a tree node to an organization tree version. This topic shows how to load organization tree nodes using HCM Data Loader.

Creating Organization Tree Nodes

This example OrganizationTreeNode.dat file loads organization tree nodes to an existing organization tree version.

```
METADATA|OrganizationTreeNode|TreeStructureCode|TreeCode|TreeVersionName|OrganizationName|
ClassificationCode|ParentOrganizationName|ParentClassificationCode
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|Vision Enterprise|
Enterprise||
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|US Division|
Division|Vision Enterprise|Enterprise
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|US LE|Legal
 Employer|US Division|Division
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|EMEA Division|
Division|Vision Enterprise|Enterprise
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|France LE|Legal
 Employer|EMEA Division|Division
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|Italy LE|Legal
 Employer|EMEA Division|Division
```

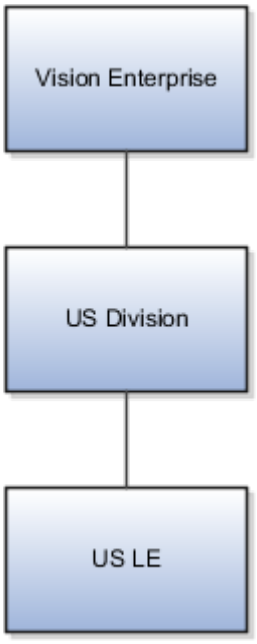
This figure shows the hierarchy nodes that this OrganizationTreeNode.dat file creates.



Deleting Organization Tree Nodes

You can delete organization tree nodes using HCM Data Loader. The organizations are removed from the tree version, but the organization objects themselves continue to exist. To delete an organization tree node and all nodes below it in the organization tree hierarchy, set the **DeleteChildNodesFlag** attribute to **Y**. This example OrganizationTreeNode.dat file deletes a node and its child nodes.

```
METADATA|OrganizationTreeNode|TreeStructureCode|TreeCode|TreeVersionName|OrganizationName|
ClassificationCode|DeleteChildNodesFlag
DELETE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|EMEA Division|
Division|Y
```

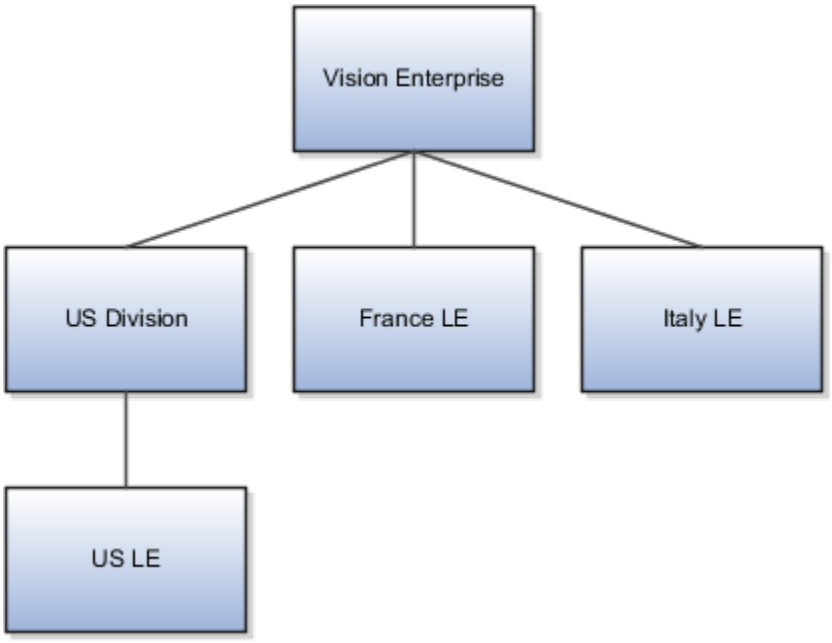


This figure shows the hierarchy after deleting the EMEA Division node and its child nodes.

To delete an organization tree node and promote its child nodes, set the **DeleteChildNodesFlag** attribute to **N**. This example OrganizationTreeNode.dat file deletes a node and promotes its child nodes.

```
METADATA|OrganizationTreeNode|TreeStructureCode|TreeCode|TreeVersionName|OrganizationName|
ClassificationCode|DeleteChildNodesFlag
DELETE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|EMEA Division|
Division|N
```

This figure shows the hierarchy after deleting the EMEA Division node and promoting its child nodes.



Related Topics

- Guidelines for Loading Organization Tree Nodes

6 Loading Market Data for Compensation

Guidelines for Loading Market Data Objects

You can load survey data from external suppliers using HCM Data Loader market data objects. The survey data can include salary, bonuses, benefits, and so on. Here are some tips for loading market data objects.

Any Type of Market Data

Before you can load any of the market data objects, you must define the survey data supplier. Let's say you're loading survey data from Hay. You need a supplier definition for Hay in your environment before you can load the data. Use the **Supplier Structures** task in the Compensation work area.

Market Data Survey Data

Here's what you need to do before you can load a **Market Data Survey Data** object. All of the tasks are in the Compensation work area.

- The survey, such as Hay Professional, using the **Surveys** task
- The value of any **CompensationTypeCode** attribute that you supply using the **Compensation Types** task
- Any of these market data job structures objects that you reference on the **Market Data Survey Data** object:
    - Market Data Career Stream
    - Market Data Career Level
    - Market Data Other Level
    - Market Data Job Family
    - Market Data Job Function

  You load just the market data job structures objects that you want to use in your compensation analyses.
- Any of these objects that you reference on the **Market Data Survey Data** object:
    - Market Data Job List
    - Market Data Location List

You need to load any job structures objects before the **Market Data Job List** object because the job list refers to the job structures objects.

Tip: To make sure that all object references resolve, load the **Market Data Survey Data** object last.

The objects referenced on the Market Data Survey Data object, such job family, can apply to more than one survey from the same supplier.

HCM Spreadsheets Data Loader Templates

You can also load market data objects using HCM Spreadsheet Data Loader. You can copy the bundled spreadsheet templates to create your own versions. You can find the bundled templates in the Compensation and Data Exchange work areas.

Reference

The Loading Market Data Using HCM Data Loader document explains how you can load data using HCM Data Loader and HCM Spreadsheet Data Loader. The document (ID 2513871.1) is on My Oracle Support at https://support.oracle.com.

Examples of Loading Market Data Segments and Survey Composites

Here are examples of how you can load market data segments and survey composites using HCM Data Loader.

Market Data Segments

Here's how you can load the market data segments using HCM Data Loader.

```
METADATA|MktSegment|SegmentId|SegmentCode|SegmentName|StatusCode|SegmentDescr|MiscText1|MiscText2|MiscText3|
SourceSystemOwner|SourceSystemId
MERGE|MktSegment||SEG889|Segment 889|A|San Mateo|ExampleABC|ExampleDEF|ExampleGHI|VISION|SEG889

METADATA|MktSegmentLocation|SegmentId|SegmentCode|LocationId|LocationCode|SetCode|SourceSystemOwner|
SourceSystemId
MERGE|MktSegmentLocation||SEG889||A1 Location|COMMON|VISION|SEG889
```

Market Data Survey Composites

Here's how you can load market data survey composites using HCM Data Loader. You can load these composites with either market segments or internal locations, but not both.

```
METADATA|MktComposite|JobId|JobCode|MarketSegmentId|MarketSegmentCode|LocationId|LocationCode|CountryCode|
CompTypeId|CompensationTypeCode|CurrencyCode|StartDate|EndDate|TenPercent|TwentyPercent|TwentyFivePercent|
ThirtyPercent|FortyPercent|FiftyPercent|SixtyPercent|SeventyPercent|SeventyFivePercent|EightyPercent|
NintyPercent|OneHundredPercent|AverageMean|SetCode

MERGE|MktComposite||JOBCD2||SEG889||A1 Location|US||BAS|USD|2001/02/02|2020/01/01|10|20|25|30|40|50|60|70|
75|80|90|100|55|Common
```

Related Topics

- Guidelines for Loading Market Data Objects

Examples of Loading Market Data Career Level, Career Stream, and Other Level

Here are examples of how you can load the Market Data Career Level, Market Data Career Stream, and Market Data Other Level objects using HCM Data Loader. A supplier of market data, such as Hay, supplies the object data.

Load Market Data Career Level

Here's how you can load three Market Data Career Level objects. The file identifies these objects using source keys.

```
METADATA|MktCareerLevel|SourceSystemOwner|SourceSystemId|VendorCode|CareerLevelCode|CareerLevelName|
Description|CareerLevelSequence|Status|CareerLevelDescr
MERGE|MktCareerLevel|VISION|CL_HAY_APPR|HAY|APPR|Apprentice|Apprentice|1|A|HAY Apprentice Career Level
 Notes.txt
MERGE|MktCareerLevel|VISION|CL_HAY_JRNY|HAY|JRNY|Journeyman|Journeyman|2|A|HAY Journeyman Career Level
 Notes.txt
MERGE|MktCareerLevel|VISION|CL_HAY_MSTR|HAY|MSTR|Master|Master|3|A|HAY Master Career Level Notes.txt
```

You include the .txt files named on the **CareerLevelDescr** attribute in the **ClobFiles** folder of the .zip file.

Load Market Data Career Stream

Here's how you can load three Market Data Career Stream objects. The file identifies these objects using source keys.

```
METADATA|MktCareerStream|SourceSystemOwner|SourceSystemId|VendorCode|CareerStreamCode|CareerStreamName|
Description|CareerStreamSequence|Status|CareerStreamDescr
MERGE|MktCareerStream|VISION|CS_HAY_CONT|HAY|CONT|Contributing|Contributing|1|A|HAY Contributing Career
 Stream Notes.txt
MERGE|MktCareerStream|VISION|CS_HAY_JOUR|HAY|JOUR|Journey|Journey|2|A|HAY Journey Career Stream Notes.txt
MERGE|MktCareerStream|VISION|CS_HAY_ADVA|HAY|ADVA|Advanced|Advanced|3|A|HAY Advanced Career Stream Notes.txt
```

You include the .txt files named on the **CareerStreamDescr** attribute in the **ClobFiles** folder of the .zip file.

Load Market Data Other Level

Here's how you can load four Market Data Other Level objects. The file identifies these objects using source keys.

```
METADATA|MktOtherLevel|SourceSystemOwner|SourceSystemId|VendorCode|OtherLevelCode|OtherLevelName|
Description|OtherLevelSequence|Status|OtherLevelDescr
MERGE|MktOtherLevel|VISION|OL_HAY_1|HAY|100|Associate|Associate|1|A|HAY Associate Other Level Notes.txt
MERGE|MktOtherLevel|VISION|OL_HAY_2|HAY|200|Analyst|Analyst|2|A|HAY Analyst Other Level Notes.txt
MERGE|MktOtherLevel|VISION|OL_HAY_3|HAY|300|Senior|Senior|3|A|HAY Senior Other Level Notes.txt
MERGE|MktOtherLevel|VISION|OL_HAY_4|HAY|400|Principal|Principal|4|A|HAY Principal Other Level Notes.txt
```

You include the .txt files named on the **OtherLevelDescr** attribute in the **ClobFiles** folder of the .zip file.

Related Topics

- Guidelines for Loading Market Data Objects

Examples of Loading Market Data Job Family and Function

Here are examples of how you can load the **Market Data Job Family** and **Market Data Job Function** objects using HCM Data Loader. A supplier of market survey data, such as Hay, supplies the object data.

Create a Market Data Job Family

Here's how you can create six **Market Data Job Family** objects. The file identifies these objects using source keys.

```
METADATA|MktJobFamily|SourceSystemOwner|SourceSystemId|VendorCode|JobFamilyCode|JobFamilyName|Description|
JobFamilySequence|Status|JobFamilyDescr
MERGE|MktJobFamily|VISION|JF_HAY_HR-BEN|HAY|HR-BEN|Benefits|Human Resources - Benefits|1|A|HAY Benefits Job
 Family Notes.txt
MERGE|MktJobFamily|VISION|JF_HAY_HR-CMP|HAY|HR-CMP|Compensation|Human Resources - Compensation|2|A|HAY
 Compensation Job Family Notes.txt
MERGE|MktJobFamily|VISION|JF_HAY_HR-DEV|HAY|HR-DEV|Organizational Development|Human Resources - OD|3|A|HAY
 OD Job Family Notes.txt
MERGE|MktJobFamily|VISION|JF_HAY_AC-REC|HAY|AC-PAY|Accounts Receivable|Accounting - Accounts Receivable|4|A|
HAY AR Job Family Notes.txt
MERGE|MktJobFamily|VISION|JF_HAY_AC-PAY|HAY|AC-PAY|Accounts Payable|Accounting - Accounts Payable|5|A|HAY AP
 Job Family Notes.txt
MERGE|MktJobFamily|VISION|JF_HAY_AC-COS|HAY|AC-COS|Cost Accounting|Accounting - Cost Accounting|6|A|HAY Cost
 Accounting Job Family Notes.txt
```

You include the .txt files named on the **JobFamilyDescr** attribute in the **ClobFiles** folder of the .zip file.

Create Market Data Job Functions

Here's how you can create three **Market Data Job Function** objects. The file identifies these objects using source keys.

```
METADATA|MktJobFunction|SourceSystemOwner|SourceSystemId|VendorCode|JobFunctionCode|JobFunctionName|
Description|JobFunctionSequence|Status|JobFunctionDescr
MERGE|MktJobFunction|VISION|JFN_HAY_ACCT|HAY|ACCT|Accounting|Accounting Jobs|1|A|HAY Accounting Job Function
 Notes.txt
MERGE|MktJobFunction|VISION|JFN_HAY_HR|HAY|HR|Human Resources|Human Resources Jobs|2|A|HAY HR Job Function
 Notes.txt
MERGE|MktJobFunction|VISION|JFN_HAY_IT|HAY|IT|Information Technology|Information Technology Jobs|3|A|HAY IT
 Job Function Notes.txt
```

You include the .txt files named on the **JobFunctionDescr** attribute in the **ClobFiles** folder of the .zip file.

Related Topics

- Guidelines for Loading Market Data Objects

Examples of Loading Market Data Location List and Job List

Here are examples of how you can load the **Market Data Location List** and **Market Data Job List** objects using HCM Data Loader. A supplier of market data, such as Hay, provides the object data.

Create a Market Data Location List

Here's how you can create five locations with location mappings. The file identifies these objects using source keys.

```
METADATA|MktLocationList|SourceSystemOwner|SourceSystemId|VendorCode|SurveyLocationCode|SurveyLocationName|
Status
MERGE|MktLocationList|VISION|LOC_HAY_AKR|HAY|AKR|Akron, OH|A
MERGE|MktLocationList|VISION|LOC_HAY_ALB|HAY|ALB|Albany, NY|A
MERGE|MktLocationList|VISION|LOC_HAY_ALL|HAY|AKR|Allentown, PA|A
MERGE|MktLocationList|VISION|LOC_HAY_ALQ|HAY|ALQ|Albuquerque, NM|A
MERGE|MktLocationList|VISION|LOC_HAY_CHI|HAY|CHI|Chicago, IL|A

METADATA|MktLocationMapping|SourceSystemOwner|SourceSystemId|SurveyLocationId(SourceSystemId)|
LocationId(SourceSystemId)|VendorCode
MERGE|MktLocationMapping|VISION|LM_HAY_AKR|LOC_HAY_AKR|HAM_OH|HAY
MERGE|MktLocationMapping|VISION|LM_HAY_ALB|LOC_HAY_ALB|ALBANY_NY|HAY
MERGE|MktLocationMapping|VISION|LM_HAY_ALL|LOC_HAY_ALL|BUTLER_PA|HAY
MERGE|MktLocationMapping|VISION|LM_HAY_ALQ|LOC_HAY_ALQ|ALB_NM|HAY
MERGE|MktLocationMapping|VISION|LM_HAY_CHI|LOC_HAY_CHI|CHI_IL|HAY
```

When multiple internal locations map to a single external location, you need to create the mapping. You can do that as part of the import or later, using the **Supplier Surveys** task in the Compensation work area. You can optionally create mappings between a single internal location and a single external location.

Create a Market Data Job List

Here's how you can create seven jobs. The file identifies these objects using source keys.

```
METADATA|MktJobList|SourceSystemOwner|SourceSystemId|VendorCode|VendorJobCode|Description|VendorJobTitle|
VendorJobDescrChar|Status|JobId(SourceSystemId)|JobFamilyId(SourceSystemId)|JobFunctionId(SourceSystemId)|
CareerLevelId(SourceSystemId)|CareerStreamId(SourceSystemId)|OtherLevelId(SourceSystemId)
MERGE|MktJobList|VISION|HAY-ACCT_CA|HAY|HAY-ACCT_CA|Cost accountant match job notes|Cost Accountant|Cost
 Accountant job|A|VIS_PAY_CA|JF_HAY_AC-COS|JFN_HAY_ACCT|||
MERGE|MktJobList|VISION|HAY-ACCT_PA|HAY|HAY-ACCT_PA||Payables Accountant|Payables Accountant job|A|
VIS_PAY_PA|JF_HAY_AC-PAY|JFN_HAY_ACCT|||
MERGE|MktJobList|VISION|HAY-ASSOC_ENG|HAY|HAY-ASSOC_ENG||Associate Engineer|Associate Engineer job|A|
VIS_IT_AENG||JFN_HAY_IT||CS_HAY_CONT|
MERGE|MktJobList|VISION|HAY-ENG|HAY|HAY-ENG||Engineer|Engineer job|A|VIS_IT_ENG||JFN_HAY_IT||CS_HAY_JOUR|
MERGE|MktJobList|VISION|HAY-PLUM|HAY|HAY-PLUM||Plumber 1|Apprentice Plumber 1 job|A||||CL_HAY_APPR||
MERGE|MktJobList|VISION|HAY-ASSOC_CONSLT|HAY|HAY-ASSOC_CONSLT||Associate Consultant|Associate Consultant
 job|A|VIS_ASSOC_CONS|||||OL_HAY_1
MERGE|MktJobList|VISION|HAY-CONSLT|HAY|HAY-CONSLT||Consultant|Consultant job|A|VIS_CONS|||||OL_HAY_2
```

You can map survey jobs to internal jobs using the **Market Data Job List** object or the **Supplier Surveys** task.

Related Topics

- Guidelines for Loading Market Data Objects

Example of Loading Market Data Survey Data

Here's an example of how you can load the **Market Data Survey Data** object using HCM Data Loader. A supplier of market survey data, such as Hay, provides the object data.

Loading Market Data Survey Data

Here's how you can load a single **Market Data Survey Data** object. It has a unique batch name, and it references jobs and locations that you defined before loading this object. The file identifies the **Market Data Job List**, **Market Data Job Location**, and **Market Data Survey Data** objects using source keys.

```
METADATA|MktSurveyData|SourceSystemOwner|SourceSystemId|VendorCode|SurveyCode|BatchName|
CompensationTypeCode|JobListId(SourceSystemId)|VendorLocationId(SourceSystemId)|Industry|TenPercent|
TwentyPercent|TwentyFivePercent|ThirtyPercent|FortyPercent|FiftyPercent|SixtyPercent|SeventyPercent|
SeventyFivePercent|EightyPercent|NintyPercent|OneHundredPercent|AverageMean|Currency|CountryCode|
ParticipatingIncumbants|ParticipatingCompanies
MERGE|MktSurveyData|VISION|HAY_ACCT_CA_CHI|HAY|2019|HAY_2019|BAS|HAY-ACCT_CA|LOC_HAY_CHI|All|10000|20000|
25000|30000|40000|55000|60000|70000|75000|80000|90000|100000|50000|USD|US|653|5069
```

Reference

The Loading Market Data Using HCM Data Loader document explains how you can load data using HCM Data Loader and HCM Spreadsheet Data Loader. The document (ID 2513871.1) is on My Oracle Support at https://support.oracle.com.

Related Topics

- Guidelines for Loading Market Data Objects

7 Loading Stock and External Data for Compensation

Example of Loading External Data

Here's an example of how you can load a single external data object from legacy applications and suppliers using HCM Data Loader. You can load any data into columns labeled 1 through 50.

Scenario

```
METADATA|ExternalWorkerData|ExternalWorkerDataId|PersonNumber|LegalEmployer|WorkerNumber|AssignmentNumber|
RecordTypeCd|StartDate|EndDate|Value1|Currency|Value2|Value3|SequenceNumber

MERGE|ExternalWorkerData||300100006125324|||Data from a third-party supplier|2020/01/01||Value1|USD|Value2|
Value3||5
```

Related Topics

- Record Type Lookups for External Compensation Data

Example of Loading Stock

Here's an example of how you can load stock data from external suppliers using HCM Data Loader.

Scenario

```
METADATA|Stock|TradingSymbol|AssignmentNumber|GrantType|AssignmentId(SourceSystemId)|SourceSystemOwner|
OriginalGrantDate|ExpirationDate
MERGE|Stock|ORCL|E955160008174705|EQ|Stock0001|VISION|2013/09/11|2014/09/09
```

Related Topics

- Compensation History Categories
- Overview of Personal Compensation Information and Contribution Management

8 Loading Progression Grade Ladders, Rates, and Rules for Compensation

Guidelines for Loading Progression Grade Ladders

You need to understand key aspects of progression grade ladders to load them successfully using HCM Data Loader. For example, how you load rates differs depending on whether your progression grade ladder has grades with steps or without steps.

Before You Load Progression Grade Ladders

These referenced objects need to exist in the destination environment:

- Grade and Grade Set
- Legislative Data Group

If you're using action reasons, you also need to define or load them before you load progression grade ladders.

Progression Grade Ladder Attribute Settings

For the Progression Grade Ladder object, set the attributes shown here:

Attribute	Without Step Value	With Step Value
GradeType	`GRADE`	`STEP`
ProgressionStyleCode	`CMP_GSP_GP`	`CMP_GSP_GSP`
ProgActionCode	CMP_GRADE_STEP_PROGRESSION	CMP_GRADE_STEP_PROGRESSION
SalaryActionCode	CMP_GRADE_STEP_PROGRESSION	CMP_GRADE_STEP_PROGRESSION

Rate Loading

Here's how you load the rates:

Grade Type	How You Load Rates
`GRADE`	Use the separate Progression Grade Rate object.
`STEP`	Use the Progression Step Rate and Progression Step Rate Value components of the Progression Grade Ladder object.

Grade Type	How You Load Rates
	Tip: Give the Progression Grade Ladder and Progression Step Rate components the same name because they've a one-to-one relationship. If you created a ladder with steps using the Progression Grade Ladder compensation task, the step rate and the ladder already have the same name.

Translatable Names

You can translate the name of the progression grade ladder using the Progression Grade Ladder Translation object.

- If you're using grades with steps, you can also translate the name of the progression step rate. Use the Progression Step Rate Translation object. If you're using the same name for the progression grade ladder and progression step rate, as recommended, then use the same translated value for both.
- If you're using grades without steps, you can also translate the name of the progression grade rate. Use the Progression Grade Rate Translation object.

Date and Salary Fast Formulas

You can optionally use fast formulas in your progression grade ladders to decide transaction dates or salary rates. If you do use them, the formulas must exist in the destination environment before you load Progression Grade Ladder objects.

- Date formulas need to have the Salary Progression Date Determination Rule type.
- Salary calculation formulas need to have the Salary Progression Rate Calculation Rule type.

When a fast formula decides the values of these date attributes, set the attributes to `CMP_GSP_DATE_RULE`:

- ProgressionDateCode
- SalaryChangeDateCode
- RateChangeDateCode

Then specify the formula name on the ProgressionDateRuleName, SalaryChangeDateRuleName, or RateChangeDateRuleName attribute, as appropriate.

When fast formula decide the value for the SalaryCalculationMethodCode attribute, set the attribute to `CMP_GSP_RATE_RULE`. Then specify the formula name with the SalaryCalculationRuleName attribute.

The fast formulas must exist in the target environment before you load Progression Grade Ladder objects. Date formulas must have the Salary Progression Date Determination Rule type. Salary calculation formulas must have the Salary Progression Rate Calculation Rule type.

Progression Grade Ladder Deletion

You can delete a progression grade ladder if another object, such as an assignment, doesn't reference the ladder. Any linked grades and eligibility profiles are unaffected.

- If the ladder doesn't have steps, you need to delete the linked Progression Grade Rate and Progression Rule objects separately.

- If the ladder has steps, when you delete the ladder you're also deleting the linked step rates and values. You need to delete the linked Progression Rule object separately.

Reference

The Loading Data for Grade Step Progression document explains how you can load and update grade ladder and grade attributes for grade step progression. The document (ID 2569900.1) is on My Oracle Support at https://support.oracle.com.

Related Topics

- Examples of Loading Progression Grade Ladders with Steps
- Examples of Loading Progression Grade Ladders Without Steps

Examples of Loading Progression Grade Ladders with Steps

Here are examples of how you can load Progression Grade Ladder objects with steps.

Create a Progression Grade Ladder with Steps Using Source Keys

This example ProgressionGradeLadder.dat file creates a progression grade ladder with steps. The file includes Progression Grade Ladder, Grades In Progression Ladder, Progression Step Rate, and Progression Step Rate Value components. All components use source keys. As recommended, the progression grade ladder and progression step rate have the same name.

This excerpt shows the Progression Grade Ladder component.

```
METADATA|ProgressionGradeLadder|SourceSystemOwner|SourceSystemId|EffectiveStartDate|GradeLadderName|
GradeType|ActiveStatus|GradeSetCode|AllowProgOverrideFlag|AllowSalaryOverrideFlag|AutoProgressionCode|
AutoSalChangeCode|ProgressionDateCode|ProgressionStyleCode|RateChangeDateCode|SalaryCalculationMethodCode|
SalaryChangeDateCode|UpdateSalaryFlag|ProgActionCode|SalaryActionCode|LegislativeDataGroupId(SourceSystemId)
MERGE|ProgressionGradeLadder|VISION|GSP_ANALYSTS_PGL|2000/02/15|GSP Analysts|STEP|A|COMMON|N|
N|CMP_GSP_PROG_MANUAL|CMP_GSP_PROG_MANUAL|CMP_GSP_PROCESS_DT|CMP_GSP_GSP|CMP_GSP_PROCESS_DT|
CMP_GSP_GRADE_STEP_RATE|CMP_GSP_ELIGIBILITY_DT|Y|CMP_GRADE_STEP_PROGRESSION|CMP_GRADE_STEP_PROGRESSION|
GSP_US_LDG1
```

This excerpt shows the Grades In Progression Ladder components.

```
METADATA|GradesInProgressionLadder|SourceSystemOwner|SourceSystemId|EffectiveStartDate|
GradesInLadderSequence|GradeSetCode|GradeId(SourceSystemId)|GradeLadderId(SourceSystemId)
MERGE|GradesInProgressionLadder|VISION|GSP_ANALYSTS_PGL_GR1|2000/02/15|1|COMMON|ANALYST1_GR|GSP_ANALYSTS_PGL
MERGE|GradesInProgressionLadder|VISION|GSP_ANALYSTS_PGL_GR2|2000/02/15|2|COMMON|ANALYST2_GR|GSP_ANALYSTS_PGL
MERGE|GradesInProgressionLadder|VISION|GSP_ANALYSTS_PGL_GR3|2000/02/15|3|COMMON|ANALYST3_GR|GSP_ANALYSTS_PGL
MERGE|GradesInProgressionLadder|VISION|GSP_ANALYSTS_PGL_GR4|2000/02/15|4|COMMON|ANALYST4_GR|GSP_ANALYSTS_PGL
```

This excerpt shows the Progression Step Rate component. To indicate that this rate is for grade step progression, the RateType value is **SALARY** and the ProgressionRateFlag attribute is **Y**.

```
METADATA|ProgressionStepRate|SourceSystemOwner|SourceSystemId|EffectiveStartDate|
LegislativeDataGroupId(SourceSystemId)|GradeLadderId(SourceSystemId)|RateName|RateType|CurrencyCode|
RateFrequency|AnnualizationFactor|ActiveStatus|ProgressionRateFlag
MERGE|ProgressionStepRate|VISION|GSP_ANALYSTS_PGL_RT|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL|GSP Analysts|
SALARY|USD|HOURLY|2080|A|Y
```

This excerpt shows the Progression Step Rate Value components.

```
METADATA|ProgressionStepRateValue|SourceSystemOwner|SourceSystemId|EffectiveStartDate|
LegislativeDataGroupId(SourceSystemId)|RateId(SourceSystemId)|RateObjectId(SourceSystemId)|
StepRateValueAmount
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR1_S1|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST1_GRS1|20.00
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR1_S2|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST1_GRS2|22.75
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR2_S1|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST2_GRS1|24.20
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR2_S2|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST2_GRS2|26.00
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR2_S3|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST2_GRS3|28.25
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR2_S4|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST2_GRS4|30.25
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR3_S1|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST3_GRS1|32.00
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR3_S2|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST3_GRS2|34.80
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR3_S3|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST3_GRS3|36.15
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR4_S1|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST4_GRS1|38.00
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR4_S2|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST4_GRS2|40.10
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR4_S3|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST4_GRS3|42.00
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR4_S4|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST4_GRS4|45.25
MERGE|ProgressionStepRateValue|VISION|GSP_ANALYSTS_PGL_GR4_S5|2000/02/15|GSP_US_LDG1|GSP_ANALYSTS_PGL_RT|
ANALYST4_GRS5|48.15
```

Create a Progression Grade Ladder with Steps Using Fast Formulas and User Keys

This example ProgressionGradeLadder.dat file creates a progression grade ladder with steps. The file includes Progression Grade Ladder, Grades In Progression Ladder, Progression Step Rate, and Progression Step Rate Value components. All components use user keys. As recommended, the progression grade ladder and progression step rate have the same name.

This excerpt shows the Progression Grade Ladder component. It includes references to four existing fast formulas on the ProgressionDateCode, RateChangeDateCode, SalaryCalculationMethodCode, and SalaryChangeDateCode attributes.

```
METADATA|ProgressionGradeLadder|EffectiveStartDate|GradeLadderName|ActionReasonCode|GradeType|
ActiveStatus|GradeSetCode|AllowProgOverrideFlag|AllowSalaryOverrideFlag|AutoProgressionCode|
AutoSalChangeCode|ProgressionDateCode|ProgressionStyleCode|RateChangeDateCode|SalaryCalculationMethodCode|
SalaryChangeDateCode|UpdateSalaryFlag|ProgActionCode|ProgActionReasonCode|SalaryActionCode|
LegislativeDataGroup|ProgressionDateRuleName|SalaryChangeDateRuleName|RateChangeDateRuleName|
SalaryCalculationRuleName
MERGE|ProgressionGradeLadder|2016/01/01|GSP Nurses|GSP_NEW|STEP|A|COMMON|N|N|CMP_GSP_PROG_MANUAL|
CMP_GSP_PROG_MANUAL|CMP_GSP_DATE_RULE|CMP_GSP_GSP|CMP_GSP_DATE_RULE|CMP_GSP_RATE_RULE|CMP_GSP_DATE_RULE|
Y|CMP_GRADE_STEP_PROGRESSION|NEW_CONTRACT|CMP_GRADE_STEP_PROGRESSION|GBI United States LDG|GSP Assignment
 Date|GSP Salary Date|GSP Rate Sync Salary Date|GSP Rate Calc
```

This excerpt shows the Grades In Progression Ladder components.

```
METADATA|GradesInProgressionLadder|EffectiveStartDate|GradesInLadderSequence|GradeSetCode|GradeCode|
GradeLadderName
MERGE|GradesInProgressionLadder|2016/01/01|1|COMMON|NURSE_1|GSP Nurses
```

```
MERGE|GradesInProgressionLadder|2016/01/01|2|COMMON|NURSE_2|GSP Nurses
MERGE|GradesInProgressionLadder|2016/01/01|3|COMMON|NURSE_3|GSP Nurses
```

This excerpt shows the Progression Step Rate component. To indicate that this rate is for grade step progression, RateType is **SALARY** and ProgressionRateFlag is **Y**.

```
METADATA|ProgressionStepRate|EffectiveStartDate|RateName|RateType|CurrencyCode|RateFrequency|
AnnualizationFactor|ActiveStatus|LegislativeDataGroup|GradeLadderName|ProgressionRateFlag
MERGE|ProgressionStepRate|2016/01/01|GSP Nurses|SALARY|USD|Hourly|2080|A|GBI United States LDG|GSP Nurses|Y
```

This excerpt shows the Progression Step Rate Value components.

```
METADATA|ProgressionStepRateValue|EffectiveStartDate|StepRateValueAmount|RateName|GradeSetCode|GradeCode|
GradeStepName|LegislativeDataGroup|GradeLadderName
MERGE|ProgressionStepRateValue|2016/01/01|20.00|GSP Nurses|COMMON|NURSE_1|Step 1|GBI United States LDG|GSP
 Nurses
MERGE|ProgressionStepRateValue|2016/01/01|22.00|GSP Nurses|COMMON|NURSE_1|Step 2|GBI United States LDG|GSP
 Nurses
MERGE|ProgressionStepRateValue|2016/01/01|24.00|GSP Nurses|COMMON|NURSE_2|Step 1|GBI United States LDG|GSP
 Nurses
MERGE|ProgressionStepRateValue|2016/01/01|26.00|GSP Nurses|COMMON|NURSE_2|Step 2|GBI United States LDG|GSP
 Nurses
MERGE|ProgressionStepRateValue|2016/01/01|28.00|GSP Nurses|COMMON|NURSE_2|Step 3|GBI United States LDG|GSP
 Nurses
MERGE|ProgressionStepRateValue|2016/01/01|30.00|GSP Nurses|COMMON|NURSE_2|Step 4|GBI United States LDG|GSP
 Nurses
MERGE|ProgressionStepRateValue|2016/01/01|32.00|GSP Nurses|COMMON|NURSE_3|Step 1|GBI United States LDG|GSP
 Nurses
MERGE|ProgressionStepRateValue|2016/01/01|34.00|GSP Nurses|COMMON|NURSE_3|Step 2|GBI United States LDG|GSP
 Nurses
MERGE|ProgressionStepRateValue|2016/01/01|36.00|GSP Nurses|COMMON|NURSE_3|Step 3|GBI United States LDG|GSP
 Nurses
```

Delete Progression Grade Ladders with Steps

This example ProgressionGradeLadder.dat file deletes the GSP Nurses progression grade ladder, which it references by its user key. This file deletes the progression grade ladder, including the step rates and values.

```
METADATA|ProgressionGradeLadder|GradeLadderName
DELETE|ProgressionGradeLadder|GSP Nurses
```

You need to delete the linked Progression Rule object separately.

Related Topics

- Guidelines for Loading Progression Grade Ladders
- Examples of Loading Progression Grade Ladders Without Steps

Examples of Loading Progression Grade Ladders Without Steps

Here are examples of how you can load Progression Grade Ladder objects without steps.

Create a Progression Grade Ladder Without Steps Using Source Keys

This example ProgressionGradeLadder.dat file creates a progression grade ladder where the grades don't have steps. The file includes Progression Grade Ladder and Grades In Progression Grade Ladder components. Both components use source keys. The rates are defined separately using Progression Grade Rate objects.

This excerpt shows the Progression Grade Ladder component.

```
METADATA|ProgressionGradeLadder|SourceSystemOwner|SourceSystemId|EffectiveStartDate|GradeLadderName|
GradeType|ActiveStatus|GradeSetCode|AllowProgOverrideFlag|AllowSalaryOverrideFlag|AutoProgressionCode|
AutoSalChangeCode|ProgressionDateCode|ProgressionStyleCode|RateChangeDateCode|SalaryCalculationMethodCode|
SalaryChangeDateCode|UpdateSalaryFlag|ProgActionCode|SalaryActionCode|LegislativeDataGroupId(SourceSystemId)
MERGE|ProgressionGradeLadder|VISION|GSP_DEVELOPMENT|2000/02/15|GSP Development|GRADE|A|COMMON|
N|N|CMP_GSP_PROG_MANUAL|CMP_GSP_PROG_MANUAL|CMP_GSP_PROCESS_DT|CMP_GSP_GP|CMP_GSP_PROCESS_DT|
CMP_GSP_GRADE_STEP_RATE|CMP_GSP_ELIGIBILITY_DT|Y|CMP_GRADE_STEP_PROGRESSION|CMP_GRADE_STEP_PROGRESSION|
GSP_US_LDG1
```

This excerpt shows the Grades In Progression Grade Ladder components.

```
METADATA|GradesInProgressionLadder|SourceSystemOwner|SourceSystemId|EffectiveStartDate|
GradesInLadderSequence|GradeSetCode|GradeId(SourceSystemId)|GradeLadderId(SourceSystemId)
MERGE|GradesInProgressionLadder|VISION|GSP_DEVELOPMENT_GIL1|2000/02/15|1|COMMON|DEV1_GSP|GSP_DEVELOPMENT
MERGE|GradesInProgressionLadder|VISION|GSP_DEVELOPMENT_GIL2|2000/02/15|2|COMMON|DEV2_GSP|GSP_DEVELOPMENT
MERGE|GradesInProgressionLadder|VISION|GSP_DEVELOPMENT_GIL3|2000/02/15|3|COMMON|DEV3_GSP|GSP_DEVELOPMENT
MERGE|GradesInProgressionLadder|VISION|GSP_DEVELOPMENT_GIL4|2000/02/15|4|COMMON|DEV4_GSP|GSP_DEVELOPMENT
MERGE|GradesInProgressionLadder|VISION|GSP_DEVELOPMENT_GIL5|2000/02/15|5|COMMON|DEV5_GSP|GSP_DEVELOPMENT
MERGE|GradesInProgressionLadder|VISION|GSP_DEVELOPMENT_GIL6|2000/02/15|6|COMMON|DEV6_GSP|GSP_DEVELOPMENT
```

Create a Progression Grade Ladder Without Steps Using User Keys

This example ProgressionGradeLadder.dat file creates a progression grade ladder where the grades don't have steps. The file includes Progression Grade Ladder and Grades In Progression Grade Ladder components, which both use user keys. The rates are defined separately using Progression Grade Rate objects.

This excerpt shows the Progression Grade Ladder component.

```
METADATA|ProgressionGradeLadder|EffectiveStartDate|GradeLadderName|GradeType|ActiveStatus|GradeSetCode|
AllowProgOverrideFlag|AllowSalaryOverrideFlag|AutoProgressionCode|AutoSalChangeCode|ProgressionDateCode|
ProgressionStyleCode|RateChangeDateCode|SalaryCalculationMethodCode|SalaryChangeDateCode|UpdateSalaryFlag|
ProgActionCode|SalaryActionCode|LegislativeDataGroup
MERGE|ProgressionGradeLadder|2010/01/01|GSP Staff|GRADE|A|COMMON|N|N|CMP_GSP_PROG_MANUAL|
CMP_GSP_PROG_MANUAL|CMP_GSP_PROCESS_DT|CMP_GSP_GP|CMP_GSP_PROCESS_DT|CMP_GSP_GRADE_STEP_RATE|
CMP_GSP_ELIGIBILITY_DT|Y|CMP_GRADE_STEP_PROGRESSION|CMP_GRADE_STEP_PROGRESSION|GBI United States LDG
```

This excerpt shows the Grades in Progression Grade Ladder components.

```
METADATA|GradesInProgressionLadder|EffectiveStartDate|GradesInLadderSequence|GradeSetCode|GradeCode|
GradeLadderName
MERGE|GradesInProgressionLadder|2010/01/01|1|COMMON|STAFF1|GSP Staff
MERGE|GradesInProgressionLadder|2010/01/01|2|COMMON|STAFF2|GSP Staff
MERGE|GradesInProgressionLadder|2010/01/01|3|COMMON|STAFF3|GSP Staff
MERGE|GradesInProgressionLadder|2010/01/01|4|COMMON|STAFF4|GSP Staff
```

Delete Progression Grade Ladders Without Steps

This example ProgressionGradeLadder.dat file deletes a progression grade ladder, which it references by its source key. This progression grade ladder includes grades without steps, so you need to delete any linked Progression Grade Rate and Progression Rule objects separately.

```
METADATA|ProgressionGradeLadder|SourceSystemOwner|SourceSystemId
DELETE|ProgressionGradeLadder|VISION|GSP_DEVELOPMENT
```

Related Topics

- Guidelines for Loading Progression Grade Ladders
- Examples of Loading Progression Grade Ladders with Steps

Guidelines for Loading Progression Grade Rates

You need to understand some key aspects of progression grade rates to load them successfully using HCM Data Loader.

CAUTION: You load progression grade rates only if your grades don't have steps.

Before You Load Progression Grade Rates

These referenced objects need to exist in the destination environment:

- Grade and Grade Set
- Legislative Data Group
- Progression Grade Ladder

Progression Grade Rate Names

When you create a progression grade ladder on the Progression Grade Ladders page, you're also creating the linked progression grade rate. The rate has the same name as the ladder.

A progression grade ladder can have only one progression grade rate, and both objects have the same legislative data group. When you create a progression grade rate using HCM Data Loader, give it the same name and LDG as the linked progression grade ladder.

Progression Grade Rates in the Application

You can create and maintain progression grade rates with the Progression Grade Ladders task in the Compensation work area. And, you can load them using HCM Data Loader.

You can also view progression grade rates using the Grade Rates task, because progression grade rates and grade rates share the same underlying tables. But, you can't use any rates created with the **Grade Rates** task in grade step progression because they're missing key values for progression processing.

Reference

The Loading Data for Grade Step Progression document explains how you can load and update grade ladder and grade attributes for grade step progression. The document (ID 2569900.1) is on My Oracle Support at https://support.oracle.com.

Related Topics

- Examples of Loading Progression Grade Rates

Examples of Loading Progression Grade Rates

This topic provides examples showing how to load Progression Grade Rate objects using HCM Data Loader.

Creating Progression Grade Rates

In both examples in this section, the progression grade rate:

- Uses grades without grade steps, and the rates are quoted as monthly amounts
- Has the same name as its associated progression grade ladder
- Has a RateType of `SALARY`, and the ProgressionRateFlag is set to `Y` to indicate that this rate is to be used for grade step progression

This example ProgressionGradeRate.dat file creates a Progression Grade Rate object using source keys.

```
METADATA|ProgressionGradeRate|SourceSystemOwner|SourceSystemId|EffectiveStartDate|GradeRateName|
LegislativeDataGroupId(SourceSystemId)|RateType|CurrencyCode|RateFrequency|ActiveStatus|AnnualizationFactor|
ProgressionRateFlag|GradeLadderId(SourceSystemId)
MERGE|ProgressionGradeRate|VISION|GSP_DEVELOPMENT_GR|2000/02/15|GSP Development|GSP_US_LDG1|SALARY|USD|
MONTHLY|A|12|Y|GSP_DEVELOPMENT
METADATA|ProgressionGradeRateValue|SourceSystemOwner|SourceSystemId|EffectiveStartDate|
LegislativeDataGroupId(SourceSystemId)|RateId(SourceSystemId)|RateObjectId(SourceSystemId)|MinimumAmount|
MaximumAmount|ValueAmount
MERGE|ProgressionGradeRateValue|VISION|GSP_DEVELOPMENT_GRV1|2000/02/15|GSP_US_LDG1|GSP_DEVELOPMENT_GR|
DEV1_GSP|3500|6500|5000
MERGE|ProgressionGradeRateValue|VISION|GSP_DEVELOPMENT_GRV2|2000/02/15|GSP_US_LDG1|GSP_DEVELOPMENT_GR|
DEV2_GSP|4500|7500|6000
MERGE|ProgressionGradeRateValue|VISION|GSP_DEVELOPMENT_GRV3|2000/02/15|GSP_US_LDG1|GSP_DEVELOPMENT_GR|
DEV3_GSP|5500|8500|7000
MERGE|ProgressionGradeRateValue|VISION|GSP_DEVELOPMENT_GRV4|2000/02/15|GSP_US_LDG1|GSP_DEVELOPMENT_GR|
DEV4_GSP|6500|9500|8000
MERGE|ProgressionGradeRateValue|VISION|GSP_DEVELOPMENT_GRV5|2000/02/15|GSP_US_LDG1|GSP_DEVELOPMENT_GR|
DEV5_GSP|7500|10500|9000
MERGE|ProgressionGradeRateValue|VISION|GSP_DEVELOPMENT_GRV6|2000/02/15|GSP_US_LDG1|GSP_DEVELOPMENT_GR|
DEV6_GSP|8500|11500|10000
```

This example ProgressionGradeRate.dat file creates a Progression Grade Rate object using user keys.

```
METADATA|ProgressionGradeRate|EffectiveStartDate|RateType|GradeRateName|GradeLadderName|CurrencyCode|
RateFrequency|AnnualizationFactor|ActiveStatus|LegislativeDataGroup|ProgressionRateFlag
MERGE|ProgressionGradeRate|2010/01/01|Salary|GSP Staff|GSP Staff|USD|Monthly|12|A|GBI United States LDG|Y
METADATA|ProgressionGradeRateValue|EffectiveStartDate|MinimumAmount|MaximumAmount|MidValueAmount|
ValueAmount|GradeRateName|SetCode|GradeCode|LegislativeDataGroup
MERGE|ProgressionGradeRateValue|2010/01/01|2000|5000|3500|3000|GSP Staff|COMMON|STAFF1|GBI United States LDG
MERGE|ProgressionGradeRateValue|2010/01/01|4000|6000|5000|4500|GSP Staff|COMMON|STAFF2|GBI United States LDG
MERGE|ProgressionGradeRateValue|2010/01/01|5000|8000|6500|5800|GSP Staff|COMMON|STAFF3|GBI United States LDG
MERGE|ProgressionGradeRateValue|2010/01/01|6000|9000|7500|6200|GSP Staff|COMMON|STAFF4|GBI United States LDG
```

Deleting Progression Grade Rates

When you delete a progression grade rate, its grade rate values are also deleted. You can delete a progression grade rate using HCM Data Loader if another object, such as a salary basis, isn't referencing it. If the progression grade rate is used in a progression grade ladder, then delete it only if you're also deleting the related progression grade ladder.

This example ProgressionGradeRate.dat file deletes a progression grade rate, which is identified using source keys.

```
METADATA|ProgressionGradeRate|SourceSystemOwner|SourceSystemId|LegislativeDataGroupId(SourceSystemId)
DELETE|ProgressionGradeRate|VISION|GSP_DEVELOPMENT_GR|GSP_US_LDG1
```

This example ProgressionGradeRate.dat file deletes a progression grade rate, which is identified using user keys.

```
METADATA|ProgressionGradeRate|LegislativeDataGroup|GradeRateName
DELETE|ProgressionGradeRate|GBI United States LDG|GSP Staff
```

Related Topics

- Guidelines for Loading Progression Grade Rates

Guidelines for Loading Progression Rules

You need to understand some key aspects of progression rules to load them successfully using HCM Data Loader.

Before You Load Progression Rules

These referenced objects need to exist in the destination environment:

- Progression Grade Ladder
- Eligibility Profile

Progression Rules and Eligibility Profiles

A progression rule is simply the name of the eligibility profile with the conditions that must be met for progression to occur. You can associate a progression rule with a progression grade ladder, grade, or grade step. The Progression Rule object loads only the association between the eligibility profile and the progression grade ladder, grade, or grade step. Multiple progression rules can exist at each level.

Eligibility profiles associated with Progression Rule objects need a usage of **Compensation** or **Global**. And, the Progression Rule object can't reference them using source keys.

Reference

The Loading Data for Grade Step Progression document explains how you can load and update grade ladder and grade attributes for grade step progression. The document (ID 2569900.1) is on My Oracle Support at https://support.oracle.com.

Related Topics

- Examples of Loading Progression Rules

Examples of Loading Progression Rules

Here are examples of how you can load Progression Rule objects using HCM Data Loader.

Create Progression Rules

Typically, you load progression rules for either each step of the progression grade ladder or each grade if your grades don't have steps. These examples show you how to load progression rules for the progression grade ladder, grade, and grade step. They include the attributes necessary to build the keys at each level and apply these rules:

- When you load a rule for the progression grade ladder, set the SetCode, GradeId, and GradeStepId or GradeStepName attributes to `#NULL`.
- When you load a rule for a particular grade, set the GradeStepId or GradeStepName attribute to `#NULL`.
- When you load a rule for a step, supply values for all attributes.

Note: If your grades don't have steps, you need to remove the GradeStepId or GradeStepName attribute from your file.

Or, you can load the progression rules for each level separately. In this case, you include only the necessary key attributes. For example, when you load progression rules for only the progression grade ladder, you remove the SetCode, GradeId, and GradeStepId or GradeStepName attributes. Similarly, when you load progression rules for only the grades, you remove the GradeStepId or GradeStepName attribute.

Note: You can't reference the eligibility profiles using source keys.

This example ProgressionRule.dat file creates progression rules at each level using source keys. Existing eligibility profiles define the criteria for progression.

```
METADATA|ProgressionRule|SourceSystemOwner|SourceSystemId|EffectiveStartDate|GradeLadderId(SourceSystemId)|
SetCode|GradeId(SourceSystemId)|GradeStepId(SourceSystemId)|ProgressionRuleName
COMMENT Rule for Progression Grade Ladder
MERGE|ProgressionRule|VISION|GSP_ANALYSTS_PGL_RULE1|2000/02/15|GSP_ANALYSTS_PGL|#NULL|#NULL|#NULL|Active
 Employees
COMMENT Rule for Grade
MERGE|ProgressionRule|VISION|GSP_ANALYSTS_PGL_RULE_G1_1|2000/02/15|GSP_ANALYSTS_PGL|COMMON|ANALYST1_GR|
#NULL|26 Weeks of Service
COMMENT Rules for Steps
MERGE|ProgressionRule|VISION|GSP_ANALYSTS_PGL_RULE_G1S1_1|2000/02/15|GSP_ANALYSTS_PGL|COMMON|ANALYST1_GR|
ANALYST1_GRS1|Completed Training
MERGE|ProgressionRule|VISION|GSP_ANALYSTS_PGL_RULE_G1S1_2|2000/02/15|GSP_ANALYSTS_PGL|COMMON|ANALYST1_GR|
ANALYST1_GRS1|Completed Training Level 2
```

This example ProgressionRule.dat file creates progression rules at each level using user keys. Existing eligibility profiles define the criteria for progression.

```
METADATA|ProgressionRule|GradeLadderName|SetCode|GradeCode|GradeStepName|ProgressionRuleName|
EffectiveStartDate
COMMENT Rule for Progression Grade Ladder
MERGE|ProgressionRule|GSP Nurses|#NULL|#NULL|#NULL|Active Employees|2016/01/01
COMMENT Rules for Grade
MERGE|ProgressionRule|GSP Nurses|COMMON|NURSE_1|#NULL|26 Weeks of Service|2016/01/01
COMMENT Rules for Steps
MERGE|ProgressionRule|GSP Nurses|COMMON|NURSE_1|Step 1|Completed Training|2016/01/01
MERGE|ProgressionRule|GSP Nurses|COMMON|NURSE_1|Step 1|Completed Training Level 2|2016/01/01
```

Delete Progression Rules

You can delete a progression rule record using HCM Data Loader. Deleting a progression rule doesn't delete the linked eligibility profiles.

This example ProgressionRule.dat file deletes three progression rules, which it references by their source keys.

```
METADATA|ProgressionRule|SourceSystemOwner|SourceSystemId
COMMENT Delete Rule for Progression Grade Ladder
DELETE|ProgressionRule|VISION|GSP_ANALYSTS_PGL_RULE1
COMMENT Delete Rule for Grade
DELETE|ProgressionRule|VISION|GSP_ANALYSTS_PGL_RULE_G1_1
COMMENT Delete Rule for Step
DELETE|ProgressionRule|VISION|GSP_ANALYSTS_PGL_RULE_G1S1_1
```

This example ProgressionRule.dat file deletes three progression rules, which it references by their user keys. It also applies these rules because you delete progression rules at all levels:

- When you delete a rule for the progression grade ladder, set the SetCode, GradeCode, and GradeStepName attributes to `#NULL`.
- When you delete a rule for a grade, set the GradeStepName attribute to `#NULL`.
- When you delete a rule for a step, supply values for all attributes.
- If your grades don't have steps, remove the GradeStepName attribute from your file.

```
METADATA|ProgressionRule|GradeLadderName|SetCode|GradeCode|GradeStepName|ProgressionRuleName
COMMENT Delete Rule for Ladder
DELETE|ProgressionRule|GSP Nurses|#NULL|#NULL|#NULL|Active Employees
COMMENT Delete Rule for Grade
DELETE|ProgressionRule|GSP Nurses|COMMON|NURSE_1|#NULL|26 Weeks of Service
COMMENT Delete Rule for Step
DELETE|ProgressionRule|GSP Nurses|COMMON|NURSE_1|Step 1|Completed Training
```

Related Topics

- Guidelines for Loading Progression Rules

9 Loading Salary Bases, Salary Records, and Salary Range Differentials for Compensation

Guidelines for Loading Salary Basis Records

You need to understand key aspects of the Salary Basis object to load salaries and salary bases successfully using HCM Data Loader.

Payroll Elements

Elements that you link to salary bases need to exist in the destination environment before you load salary bases. And, the elements need to be configured for use with salary bases and salaries. Otherwise, your salary bases might not load successfully. For example, the salary bases might not link to a legislative data group or a payroll element.

You can manually correct salary bases using the Salary Basis task in the Compensation work area. For more information about creating the appropriate elements, see *DOC ID 1589502.1*

Grade Rates

To validate people's salaries against salary ranges defined for a grade you can link a grade rate to the salary basis. This linking also let's you display salary ranges and calculate analytical values, such as compa-ratio and range position. Here are the conditions needed for this to work:

- The person's grade needs to be in the grade rate linked to their salary basis.
- The grade rate currency needs to match the currency of the payroll element linked to the salary basis.
- The grade needs to be in the same legislative data group as the salary basis.

Salary Basis Code

To record the frequency that base pay is stored and displayed at, use the SalaryBasisCode attribute. Valid values are in the CMP_SALARY_BASIS lookup type. The PERIOD (payroll period) value means that the salary frequency comes from the payroll defined at the assignment level.

When you set Salary Basis Code to **PERIOD**, leave SalaryAnnualizationFactor blank.

Salary Basis Components

To specify whether you're loading salary basis components to itemize salary adjustments, use the ComponentUsage attribute. Valid values are in the CMP_COMPONENT_USAGE lookup type and described here.

Value	Meaning
NO_COMPONENT	Components aren't used and aren't loaded with the Salary Basis object.
SELECTED_COMPONENTS	Components are used and are loaded with the Salary Basis object.

Value	Meaning
USER_DECIDES_USAGE	Managers select the components to use when adjusting salary. Components aren't loaded with the Salary Basis object.

If you set ComponentUsage to **SELECTED_COMPONENTS**, load the components using the SalaryBasisComponent discriminator. If you omit ComponentUsage, the loader assumes NO_COMPONENT.

Tip: If you're unsure about component usage, specify **USER_DECIDES_USAGE** because it's the least restrictive.

You need to provide one Salary Basis Component record for each component type. To identify the component type, such as PROMOTIONAL, COST_OF_LIVING, or ADJUSTMENT, use the AttributeValue attribute. Valid values are in the CMP_SALARY_COMPONENTS lookup type. Use the Manage Base Pay Lookups task in the Setup and Maintenance work area to manage the lookup type.

Salary Basis Deletions

You can delete a Salary Basis object using HCM Data Loader only if the object is unassigned and wasn't used to create salary records. You can delete individual salary bases most easily using the **Salary Basis** task in the Compensation work area.

Related Topics

- Guidelines for Loading Salary Records
- Examples of Loading Salary Records

Simple Component Lookups to Use When Creating Salary Bases

You use these lookup codes when you create salary bases with simple components using the Salary Basis business object and HCM Data Loader.

HCM Data Loader Attribute	Lookup Type	Lookup Code
ComponentType	ORA_CMP_SIMPLE_COMPONENT_TYPES	ORA_USER_ENTERED_AMT ORA_USER_ENTERED_PERCENTAGE ORA_FIXED_PERCENTAGE ORA_FIXED_AMOUNT ORA_GSP_RATE_AMOUNT ORA_OVERALL_SALARY
SimpleComponentCode	ORA_CMP_SIMPLE_SALARY_COMPS	ORA_WAGE_PROGRESSION_RATE ORA_HOUSING_ALLOWANCE

HCM Data Loader Attribute	Lookup Type	Lookup Code
		ORA_CITY_ALLOWANCE ORA_TRANSPORT_ALLOWANCE ORA_VAR_ALLOWANCE ORA_TRAVEL_ALLOWANCE ORA_BASIC ORA_SPECIAL_ALLOWANCE ORA_OVERALL_SALARY
OverallSalaryImpact	ORA_CMP_AFFECT_OVERALL_SALARY	ORA_ADD_TO_SALARY ORA_NO_EFFECT ORA_SUBTRACT_TO_SALARY

Create a Salary Basis with Simple Components

This example SalaryBasis.dat file creates a Salary Basis object with simple components.

```
METADATA|SalaryBasis|ComponentUsage|SalaryBasisCode|SalaryAnnualizationFactor|SalaryBasisName|
LegislativeDataGroupName|ElementName|InputValueName|SalaryBasisType|Status|CurrencyCode
MERGE|SalaryBasis|SELECTED_COMPONENTS|ANNUAL|1|RJ Type 4 HDL1|GBI United States LDG|RJXPA10_PrimaryAmount|
Amount|ORA_SMIPLE_COMPONENTS|A|USD
METADATA|SalaryBasisSimpleComponent|BasedOnComponentCode|BasisSimpleComponentId|ComponentType|
DefaultValue|DisplaySequence|OverallSalaryImpact|ProcessingSequence|SalaryBasisId|SimpleComponentCode|
UserSelectedComponent|SalayBasisName
MERGE|SalaryBasisSimpleComponent|||ORA_USER_ENTERED_AMT||10|ORA_ADD_TO_SALARY|10||ORA_BASIC|ORA_NO|RJ Type 4
 HDL1
MERGE|SalaryBasisSimpleComponent|||ORA_OVERALL_SALARY||20|ORA_NO_EFFECT|20||ORA_OVERALL_SALARY|ORA_NO|RJ
 Type 4 HDL1
```

Guidelines for Loading Salary Records

You need to understand key aspects of salaries to load salary records successfully using HCM Data Loader.

Impact on Element Entries

When the load process creates or updates a salary record, it adds an element entry to the payroll element linked to the salary basis. When the process updates a salary record, it also end dates the previous element entry. When the process deletes a salary record, it deletes the element entry linked to the salary basis. It also removes the end date of the previous element entry.

The source of element entries created by the load process is SP (salary proposal) for salaries with these salary basis types:

- Salary amount is determined by user
- Salary amount is determined by incremental components
- Salary amount is determined by simple components

Before You Load Salary Records

You need to make sure of these configuration conditions before you can load **Salary** objects:

- Employment records must exist in the target environment.
- People's work assignments need to include a grade, and the grade needs to be included in the grade rate that's linked to each person's salary basis. These conditions are what let the salary record show the salary range and correctly calculate salary metrics.
- People with salary bases that have a frequency of PERIOD need to be assigned to a payroll defined at the payroll terms level.

Salary

You supply salary data for processing by HCM Data Loader using the Salary.dat file. When supplying data for a salary component, salary pay rate component, or salary simple component, you need to also supply the parent salary data. Here's what you need to do if you're loading salary components:

- Set the MultipleComponents attribute to **Y**.
- Set the SalaryAmount attribute to the previous base salary amount plus the sum of any salary components.

For terminated assignments, set the Date To attribute only when the salary to date is earlier than the assignment end date. Otherwise, set the salary end date to the assignment end date, and not the date in the linked payroll element.

Set the ActionCode attribute to the action linked to the salary record. Here's a list of sample values:

Action Code	Action Name
ALLOCATE_GRP_CMP	Allocate Workforce Compensation
CHANGE_SALARY	Change Salary
GLB_TRANSFER	Global Transfer
HIRE	Hire an Employee
HIRE_ADD_WORK_RELATION	Add Employee Work Relationship
PROMOTION	Promotion
REHIRE	Rehire of Worker
TRANSFER	Transfer

To create salary at the assignment level, supply a unique reference to the assignment. The Grade and FTE values are obtained from each assignment.

Salary Components

Set the ComponentReasonCode attribute. Valid values are in the CMP_SALARY_COMPONENTS lookup type. Use the Manage Base Pay Lookups task in the Setup and Maintenance work area to manage the lookup type. You can use each component reason code only once for each salary record.

The first salary record that you load needs to have the SalaryAmount attribute value equaling the sum of the incremental component values. For later salary adjustments to account for changes, the sum of the component amounts plus the previous salary amount need to equal the new salary amount. When you're not increasing or decreasing the new salary amount beyond the FTE change, enter 0 for the component change amounts. But when you're adjusting salary to account for FTE changes and an additional increase or decrease, you need to provide non-0 change amounts. Also, enter the new salary amount that accounts for the FTE change and 0 for the adjusted salary amount. For example, the current FTE is 1 and the salary is $100. As of January 1 of the next year, the FTE changes to 0.8, which means the new salary should be $80.

Specify a ChangeAmount value for each component record. You add the ChangeAmount value to the base pay amount to derive the new salary amount for the salary record. The corresponding percentage value is calculated automatically.

- To load the new salary amount adjusted for FTE, here's what you load:
  - $0 for the component amount
  - $80 for the salary amount
- To reduce or increase the new salary amount beyond $80, you need to provide the change amount for the relevant components. You also need to provide the new salary amount.

Salary Record Updates

When you update an open-ended salary, the existing salary is automatically end-dated on the day before the **From Date** value supplied for the new salary.

Related Topics

- Examples of Loading Salary Records
- Guidelines for Loading Salary Basis Records
- Can I load salary data or prevent loading when salary changes are pending?

Examples of Loading Salary Records

Here are examples of how you can load and manage salary records for a person's work assignment using HCM Data Loader.

Create a Salary Using Source Keys

This example **Salary.dat** file creates a **Salary** object with components for an existing assignment. The file identifies both the **Salary** object and the linked assignment by their source keys.

```
METADATA|Salary|SourceSystemOwner|SourceSystemId|AssignmentId(SourceSystemId)|DateFrom|SalaryAmount|
SalaryBasisName|ActionCode|SalaryApproved|MultipleComponents
MERGE|Salary|VISION|1012SAL15|1012_EMP_ASG|2015/01/01|65000|SalBasis_1007|CHANGE_SALARY|Y|N
```

```
METADATA|SalaryComponent|SourceSystemOwner|SourceSystemId|SalaryId(SourceSystemId)|DateFrom|
ComponentReasonCode|ChangeAmount|ComponentApproved
MERGE|SalaryComponent|VISION|1012SAL15_COL|1012SAL15|2015/01/01|COST_OF_LIVING|60000|Y
MERGE|SalaryComponent|VISION|1012SAL15_ADJ|1012SAL15|2015/01/01|ADJUSTMENT|5000|Y
```

Update a Salary Using Source Keys

This example **Salary.dat** file updates an existing, open-ended salary record by supplying changes to the Salary object and its components. The file identifies all components by their source keys.

```
METADATA|Salary|SourceSystemOwner|SourceSystemId|AssignmentId(SourceSystemId)|DateFrom|DateTo|SalaryAmount|
SalaryBasisName|ActionCode|SalaryApproved|MultipleComponents
MERGE|Salary|VISION|1012SAL16|1012_EMP_ASG|2016/01/01||70000|SalBasis_1007|CHANGE_SALARY|Y|N
METADATA|SalaryComponent|SourceSystemOwner|SourceSystemId|SalaryId(SourceSystemId)|DateFrom|
ComponentReasonCode|ChangeAmount|ComponentApproved
MERGE|SalaryComponent|VISION|1012SAL16_COL|1012SAL16|2016/01/01|COST_OF_LIVING|5000|Y
MERGE|SalaryComponent|VISION|1012SAL16_ADJ|1012SAL16|2016/01/01|ADJUSTMENT|0|Y
```

Create a Salary Using User Keys

This example **Salary.dat** file creates a **Salary** object with components, for an assignment. The file identifies both the **Salary** object and the linked assignment using user keys.

```
METADATA|Salary|AssignmentNumber|DateFrom|SalaryAmount|SalaryBasisName|ActionCode|SalaryApproved|
MultipleComponents
MERGE|Salary|1014|2015/01/01|53500|SalBasis_1007|CHANGE_SALARY|Y|N
METADATA|SalaryComponent|AssignmentNumber|DateFrom|ComponentReasonCode|ChangeAmount|ComponentApproved
MERGE|SalaryComponent|1014|2015/01/01|COST_OF_LIVING|52000|Y
MERGE|SalaryComponent|1014|2015/01/01|ADJUSTMENT|1500|Y
```

Update a Salary Using User Keys

This example **Salary.dat** file updates an existing, open-ended salary record by supplying changes to the **Salary** object and its components. The file identifies all components using user keys.

```
METADATA|Salary|AssignmentNumber|DateFrom|DateTo|SalaryAmount|SalaryBasisName|ActionCode|SalaryApproved|
MultipleComponents
MERGE|Salary|1014|2016/01/01||53900|SalBasis_1007|CHANGE_SALARY|Y|N
METADATA|SalaryComponent|AssignmentNumber|DateFrom|ComponentReasonCode|ChangeAmount|ComponentApproved
MERGE|SalaryComponent|1014|2016/01/01|COST_OF_LIVING|300|Y
MERGE|SalaryComponent|1014|2016/01/01|ADJUSTMENT|100|Y
```

Delete a Salary

Delete salary records using HCM Data Loader only if many records were created or loaded incorrectly and need to be deleted in bulk. Unless you're deleting all salary records for an assignment, you can delete only the salary record with the most recent salary start date. For example, if a salary record has salary start dates of January 1, March 1, and September 1 then you can delete only the September 1 record. Or, you can delete all three records.

This example **Salary.dat** file deletes an existing salary record. The file identifies the **Salary** object by its source key.

```
METADATA|Salary|SourceSystemOwner|SourceSystemId|AssignmentId(SourceSystemId)|DateFrom
DELETE|Salary|VISION|1012SAL16|EJ1012_EMP_ASG|2016/01/01
```

This example **Salary.dat** file deletes an existing salary. The file identifies the **Salary** object by its user key.

```
METADATA|Salary|AssignmentNumber|DateFrom
DELETE|Salary|1014|2016/01/01
```

Purge Salaries for an Assignment

Purge all the salary records for an assignment in bulk using HCM Data Loader and the **PurgeAssignmentSalary** attribute. You supply the delete command for the salary record with the most recent salary start date. For example, if the assignment has salary records with salary start dates of January 1, March 1, and September 1, you use the delete command with the September 1 salary. When you load the delete command, all three of salary records get purged.

This example **Salary.dat** file purges all the existing salary records of the assignment. The file identifies the **Salary** object by the assignment number.

```
METADATA|Salary|AssignmentNumber|PurgeAssignmentSalary|DateFrom
DELETE|Salary|E966169008889157|Y|2021/01/01
```

Related Topics

- Guidelines for Loading Salary Records
- Guidelines for Loading Salary Basis Records

Create a Salary with Simple Components

This example Salary.dat file creates a Salary object with simple components for an existing assignment.e

```
|COMMENT Data for Business Object: Salary Version: V2 Created on 11-11-2020
METADATA|Salary|DateFrom|SalaryAmount|MultipleComponents|AssignmentNumber|SalaryBasisName|ActionCode
MERGE|Salary|2011/01/01|10000|Y|E966169008889140|RJ Type 4 HDL1|CHANGE_SALARY
METADATA|SalarySimpleComponent|ComponentCode|Amount|Percentage|SalaryDateFrom|AssignmentNumber
MERGE|SalarySimpleComponent|ORA_BASIC|10000||2011/01/01|E966169008889140
```

Can HCM Data Loader round salary amounts when it loads salary information?

No. The data loader loads the amounts that are in the file.

Here are the three conditions when salary amounts get rounded:

- You increase the amount using a percentage, such as 3.5%.
- Salary calculations convert the amount to a different frequency, such as from monthly to annual or annual to monthly.
- You prorate the amount according to a specific full-time equivalent (FTE), such as from 1 FTE to 0.5 FTE.

Can I load salary data or prevent loading when salary changes are pending?

Yes. You can do it using the **HR_DISABL_PENDING_APPROVALS_CHECK_IN_ HCM_DATA_LOADER** profile option. To enable loading, at the **Site** level, set **Profile Value** to **No**. To prevent loading, set the **Profile Value** to **Yes**.

Why can't I delete salary records using HCM Spreadsheet Data Loader?

You can delete only the latest salary record when you use the supported **Delete only** action. Also, you need to have a **Compensation Manager** or **HR Specialist** type of role that automatically inherits the required **ORA_CMP_DELETE_WORKER_LATEST_SALARY** data privilege.

If you have a **Line Manager** type role, you don't have the necessary data privilege.

Before trying to deleting salary using the loader with session user security, sign in to the application with the same credentials. Then, try deleting the salary record from the compensation page. If you can't delete the record from the page, then you can't delete it using the loader.

Guidelines for Making Date-Effective Changes to Salary Range Differentials

The **Salary Range Differential** object is date-effective to maintain a history of any changes to the object and its child components. For example, people can see the date-effective history of a differential profile in the Compensation work area.

Here's how date-effective records are managed for **Salary Range Differential** objects when you use HCM Data Loader.

Date-Effective Records Creation

When you create or update a differential value, you're also adding a date-effective record to the differential value. You also need to add a date-effective record to the parent differential profile if it has no date-effective record for that effective start date. To do this, you need to include in the .dat file the parent Salary Range Differential component. It has to have the same effective start date as the child differential value.

Here's an example of history of changes to a differential profile.

RangeDiffId	EffectiveStartDa	EffectiveEndDat	Status	Criteria	LegislativeData(	Code	Name
123	01-01-1951	01-05-2016	Active	Location	US1	US SRD	US Range

RangeDiffId	EffectiveStartDa	EffectiveEndDat	Status	Criteria	LegislativeData(	Code	Name
123	02-05-2016	31-12-4712	Active	Location	US1	US SRD	US Range

Here's an example of the related changes to its differential values.

RangeDiffValueId	EffectiveStartDate	EffectiveEndDate	Differential	LocationId
234	01-01-1951	31-12-4712	1.3	A
345	02-05-2016	31-12-4712	0.8	B
456	02-05-2016	31-12-4712	0.9	C

When the location B value was created, a date-effective record with the location B effective start date was also added to the differential profile. But, when the location C value was created, no date-effective record was added to the differential profile. A record for that effective start date already existed in the differential profile, so no additional record was needed. If you create or update multiple differential values on the same date, then only one date-effective record is added to the parent differential profile.

> **Note:** You don't have to include the parent **Salary Range Differential** component in the .dat file if the file already has a date-effective record with the required effective start date.

Salary Range Differential Component

If you correct or update the parent Salary Range Differential object, then no date-effective record is added to its child components. For example, if you make the differential profile inactive, then a new date-effective record is added to the parent component. However, no date-effective record is added to the differential values.

Related Topics

- Examples of Loading Salary Range Differential Profiles and Values

Examples of Loading Salary Range Differential Profiles and Values

Here's an example of how you can load differential profiles and their differential values using HCM Data Loader.

Create Salary Range Differential Profiles and Values

To create a differential profile, you load the **Salary Range Differential** object and one or more **Salary Range Differential Value** components. Each **Salary Range Differential Value** component is for a single location.

CAUTION: Differential profiles reference legislative data groups, and their differential values reference locations. Referenced legislative data groups and locations need to exist in the destination environment before you load **Salary Range Differential** objects.

This example **SalaryRangeDifferential.dat** file creates a differential profile with a single differential value. The file identifies both components using source keys.

```
METADATA|SalaryRangeDifferential|EffectiveStartDate|Name|Code|Status|Criteria|SourceSystemId|
SourceSystemOwner|LegislativeDataGroupName
MERGE|SalaryRangeDifferential|2000/01/01|SalRangeDiffName333|SalRangeDiffCode333|ACTIVE|LOCATION|
SalRangeDiffSSID_333|VISION|PM DE LDG
METADATA|SalaryRangeDifferentialValue|EffectiveStartDate|SourceSystemId|SourceSystemOwner|
RangeDiffId(SourceSystemId)|LocationCode|LocationSetCode|Differential|Criteria
MERGE|SalaryRangeDifferentialValue|2000/01/01|SalRangeDiffVal_SSID_333|VISION|SalRangeDiffSSID_333|Atlanta|
Vision Corporation 202|0.2|Location
```

This example **SalaryRangeDifferential.dat** file creates a differential profile with a single differential value. The file identifies both components using user keys.

```
METADATA|SalaryRangeDifferential|EffectiveStartDate|Name|Code|Status|Criteria|LegislativeDataGroupName
MERGE|SalaryRangeDifferential|2000/01/01|SalRangeDiffName444|SalRangeDiffCode444|ACTIVE|LOCATION|PM DE LDG
METADATA|SalaryRangeDifferentialValue|EffectiveStartDate|RangeDiffCode|LocationCode|LocationSetCode|
Differential|Criteria
MERGE|SalaryRangeDifferentialValue|2000/01/01|SalRangeDiffCode444|Atlanta|Vision Corporation 202|0.2|
Location
```

Related Topics

- Guidelines for Making Date-Effective Changes to Salary Range Differentials

10 Loading Work Patterns for Workforce Scheduling

Overview of Loading Work Patterns

Work patterns define the worker's schedule pattern defined by the employment contract. You can load and assign work patterns to workers using the WorkPatterns.dat file, worker assignment numbers, and HCM Data Loader.

Before you load work patterns, you need to define shifts, work pattern types, and if you use them, work pattern templates.

Related Topics

- Shifts in Workforce Scheduling
- Work Pattern Types in Workforce Scheduling
- Work Pattern Templates in Workforce Scheduling

Example of Creating a Work Pattern Using a Template

This example WorkPattern.dat file creates a Work Pattern object using a work pattern template and the worker assignment number.

```
METADATA|WorkPattern|WorkPatternTypeName|RepeatNumber|RepeatCycle|DateFrom|DateTo|AssignmentNumber|
WorkPatternAltCode|TemplateName
MERGE|WorkPattern|WP_START_END_TIME_TYPE|2|Weeks|2023/07/01|2023/08/27|Worker01AsgNo01|WPC-300|TemplateKK
```

Example of Creating a Work Pattern Without a Template

This example WorkPattern.dat file creates a Work Pattern object using the worker assignment number. It also creates the child WorkPatternShift and WorkPatternBreak objects.

```
COMMENT Work Pattern
METADATA|WorkPattern|WorkPatternTypeName|RepeatNumber|RepeatCycle|DateFrom|DateTo|AssignmentNumber|
WorkPatternAltCode
MERGE|WorkPattern|WP_START_END_TIME_TYPE|2|Weeks|2023/07/01|2023/08/31|Worker02AsgNo01|WPC-301

COMMENT Work Pattern Shift
METADATA|WorkPatternShift|DayOfWorkPattern|ShiftStartTime|ShiftEndTime|DurationMinutes|
UnpaidBreakDurationMinutes|WorkPatternShiftCode|WorkPatternAltCode
MERGE|WorkPatternShift|4|06:00|09:00||15|WPS-200S|WPC-301

COMMENT Work Pattern Break
METADATA|WorkPatternBreak|StartTime|EndTime|BreakName|DayOfWorkPattern|ShiftStartTime|WorkPatternBreakCode|
WorkPatternShiftCode
MERGE|WorkPatternBreak|07:15|07:30|TeaBreak|4|06:00|WPBrk_Cd1|WPS-200S
```

Example of Loading Work Patterns Using Source Keys

This example WorkPattern.dat file creates a WorkPattern object using its source key and the worker assignment number. It also creates the child WorkPatternShift and WorkPatternBreak objects using their source keys.

```
COMMENT Work Pattern
METADATA|WorkPattern|WorkPatternTypeName|RepeatNumber|RepeatCycle|DateFrom|DateTo|AssignmentNumber|
WorkPatternAltCode|SourceSystemId|SourceSystemOwner
MERGE|WorkPattern|WP_START_END_TIME_TYPE|2|Weeks|2023/08/01|2023/08/27|Worker03AsgNo03|WPC-302|
WRKPAT_ASG_100|HCMWP-001

COMMENT Work Pattern Shift
METADATA|WorkPatternShift|DayOfWorkPattern|ShiftStartTime|ShiftEndTime|DurationMinutes|
UnpaidBreakDurationMinutes|WorkPatternShiftCode|SourceSystemId|WorkPatternAssignmentId(SourceSystemId)|
SourceSystemOwner
MERGE|WorkPatternShift|4|06:00|09:00||15|WPS-203|WRKPAT_SHIFT_100|WRKPAT_ASG_100|HCMWP-001

COMMENT Work Pattern Break
METADATA|WorkPatternBreak|StartTime|EndTime|BreakName|WorkPatternBreakCode|
WorkPatternShiftId(SourceSystemId)|SourceSystemId|SourceSystemOwner
MERGE|WorkPatternBreak|06:15|06:30|Tea 2|WPBrk_Cd4|WRKPAT_SHIFT_100|WRKPAT_BRK_100|HCMWP-001
```

Example of Loading Work Patterns Using GUIDs

This example WorkPattern.dat file creates a WorkPattern object using its GUID and the worker assignment number. It also creates the child WorkPatternShift and WorkPatternBreak objects using their GUIDs.

```
METADATA|WorkPattern|WorkPatternTypeName|RepeatNumber|RepeatCycle|DateFrom|DateTo|AssignmentId|
AssignmentNumber|WorkPatternAltCode|TemplateName|SourceSystemOwner|SourceSystemId|GUID
MERGE|WorkPattern|WP_FDT_TYPE|||2023/05/01|2023/06/15||Worker03AsgNo03|FTWP_003_001||HCMWP-002||
FF6AFF2DF736D165E0533D6015ACA633

METADATA|WorkPatternShift|ShiftId|ShiftName|WorkPatternShiftCode|DayOfWorkPattern|ShiftStartTime|
ShiftEndTime|DurationMinutes|UnpaidBreakDurationMinutes|WorkPatternAssignmentId(SourceSystemId)|
SourceSystemOwner|SourceSystemId|GUID|WorkPatternShiftId|WorkPatternAssignmentId(GUID)
MERGE|WorkPatternShift||||1|||630|30||HCMWP-002||FF6AD861ABCED14AE0533D6015ACCFBC||
FF6AFF2DF736D165E0533D6015ACA633
MERGE|WorkPatternShift||||2|||630|30||HCMWP-002||FF6AFF2DF733D165E0533D6015ACA633||
FF6AFF2DF736D165E0533D6015ACA633
MERGE|WorkPatternShift||||3|||630|30||HCMWP-002||FF6AFF2DF734D165E0533D6015ACA633||
FF6AFF2DF736D165E0533D6015ACA633
MERGE|WorkPatternShift||||4|||630|30||HCMWP-002||FF6AFF2DF735D165E0533D6015ACA633||
FF6AFF2DF736D165E0533D6015ACA633

METADATA|WorkPatternBreak|WorkPatternBreakCode|StartTime|EndTime|BreakDurationMinutes|BreakName|
WorkPatternShiftId(SourceSystemId)|SourceSystemOwner|SourceSystemId|GUID|WorkPatternBreakId|
WorkPatternShiftId(GUID)
MERGE|WorkPatternBreak|||||Lunch||HCMWP-002||FF6A8848FBF9CFE9E0533D6015ACCB06||
FF6AD861ABCED14AE0533D6015ACCFBC
MERGE|WorkPatternBreak|||||Lunch||HCMWP-002||FF6A8848FBFACFE9E0533D6015ACCB06||
FF6AFF2DF733D165E0533D6015ACA633
MERGE|WorkPatternBreak|||||Lunch||HCMWP-002||FF6A9294EA47D085E0533D6015ACFBC8||
FF6AFF2DF734D165E0533D6015ACA633
MERGE|WorkPatternBreak|||||Lunch||HCMWP-002||FF6A9294EA48D085E0533D6015ACFBC8||
FF6AFF2DF735D165E0533D6015ACA633
```

Example of Loading Work Patterns Using Reference Object Source Keys

Referenced objects for a work pattern can be a worker assignment or an enterprise shift. This example WorkPattern.dat file creates a WorkPattern object using the source key for the worker assignment.

It also creates the child WorkPatternShift and WorkPatternBreak objects using the source key for the enterprise shift.

```
METADATA|WorkPattern|SourceSystemOwner|SourceSystemId|AssignmentId(SourceSystemOwner)|
AssignmentId(SourceSystemId)|WorkPatternAltCode|DateFrom|DateTo|WorkPatternTypeName|RepeatNumber|RepeatCycle
MERGE|WorkPattern|HCMWP-003|WP_RTSSID_001A|FUSION|100000016657062||2023/04/01|2023/07/01|
WP_START_END_TIME_TYPE|1|Weeks

METADATA|WorkPatternShift|SourceSystemOwner|SourceSystemId|WorkPatternAssignmentId(SourceSystemId)|
WorkPatternShiftCode|DayOfWorkPattern|ShiftStartTime|ShiftEndTime|DurationMinutes|
UnpaidBreakDurationMinutes|ShiftId(SourceSystemId)|ShiftId(SourceSystemOwner)
MERGE|WorkPatternShift|HCMWP-003|WPS_RTSSID_001A|WP_RTSSID_001A||1|||||300100577492880|FUSION
MERGE|WorkPatternShift|HCMWP-003|WPS_RTSSID_002A|WP_RTSSID_001A||2|||||300100577492880|FUSION
MERGE|WorkPatternShift|HCMWP-003|WPS_RTSSID_003A|WP_RTSSID_001A||3|||||300100577492880|FUSION
MERGE|WorkPatternShift|HCMWP-003|WPS_RTSSID_004A|WP_RTSSID_001A||4|||||300100577492880|FUSION
MERGE|WorkPatternShift|HCMWP-003|WPS_RTSSID_005A|WP_RTSSID_001A||5|||||300100577492880|FUSION

METADATA|WorkPatternBreak|SourceSystemOwner|SourceSystemId|WorkPatternShiftId(SourceSystemId)|
WorkPatternBreakCode|StartTime|EndTime|BreakName
MERGE|WorkPatternBreak|HCMWP-003|WPB_RTSSID_001A|WPS_RTSSID_001A||22:30||Snacks_Brk1
MERGE|WorkPatternBreak|HCMWP-003|WPB_RTSSID_002A|WPS_RTSSID_002A||22:30||Snacks_Brk1
MERGE|WorkPatternBreak|HCMWP-003|WPB_RTSSID_003A|WPS_RTSSID_003A||22:30||Snacks_Brk1
MERGE|WorkPatternBreak|HCMWP-003|WPB_RTSSID_004A|WPS_RTSSID_004A||22:30||Snacks_Brk1
MERGE|WorkPatternBreak|HCMWP-003|WPB_RTSSID_005A|WPS_RTSSID_005A||22:30||Snacks_Brk1
```

Example of Deleting Work Patterns

You can also delete work patterns that are assigned to workers. You can have worker assignments with no work patterns assigned.

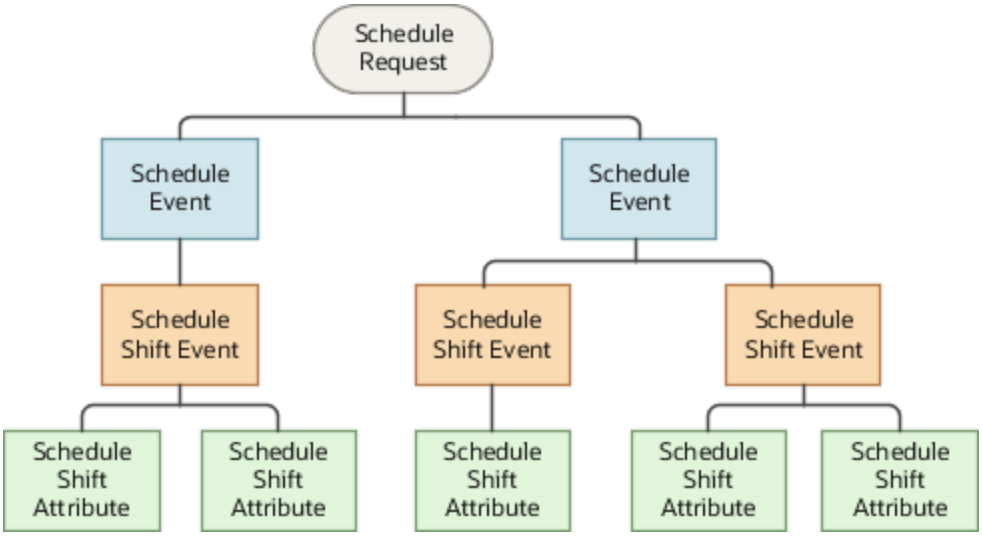
This example WorkPattern.dat file deletes a Work Pattern object using the worker assignment number. It also deletes the child WorkPatternShift and WorkPatternBreak objects.

```
COMMENT Work Pattern
METADATA|WorkPattern|WorkPatternTypeName|RepeatNumber|RepeatCycle|DateFrom|DateTo|AssignmentNumber|
WorkPatternAltCode
DELETE|WorkPattern|WP_START_END_TIME_TYPE|2|Weeks|2023/07/01|2023/08/31|Worker02AsgNo01|WPC-301

COMMENT Work Pattern Shift
METADATA|WorkPatternShift|DayOfWorkPattern|ShiftStartTime|ShiftEndTime|DurationMinutes|
UnpaidBreakDurationMinutes|WorkPatternShiftCode|WorkPatternAltCode
DELETE|WorkPatternShift|4|06:00|09:00||15|WPS-200S|WPC-301

COMMENT Work Pattern Break
METADATA|WorkPatternBreak|StartTime|EndTime|BreakName|DayOfWorkPattern|ShiftStartTime|WorkPatternBreakCode|
WorkPatternShiftCode
DELETE|WorkPatternBreak|06:15|06:30|TeaBreak|4|06:00|WPBrk_Cd1|WPS-200S
```

11 Loading Third-Party Schedules

Overview of Importing Third-Party Schedules

You can import up to four levels of schedule business objects. Managers view and optionally edit these imported objects the same way they manage their WFM schedules. Here's the hierarchy of the objects in the import file.



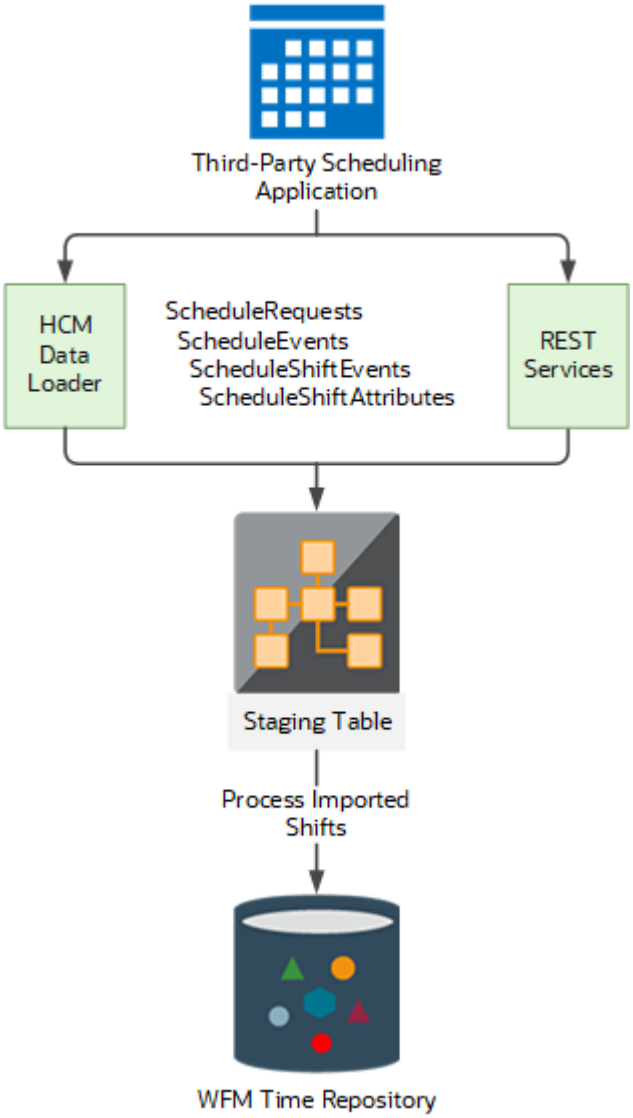
Here are import scenarios for each schedule business object and level.

Import Scenario	Affected Schedule Object	Level
Import full schedules from January 1 to 7 for Chris's team.	Schedule Request	1
Import Leslie's full schedule from January 1 to 7.	Schedule Event	2
Import Leslie's shift on January 5.	Schedule Shift Event	3
Import time attributes related to Leslie's shift on January 5.	Schedule Shift Attribute	4

Basic Import Process

Here's the basic process for how you Import third-party schedule data.

1. Import the schedules to the staging table using either REST services or HCM Data Loader.
2. Load the imported shifts to the planned schedule table of the time repository using the Scheduled Processes task and Process Imported Shifts process.



Related Topics

- How You Plan and Publish WFM Schedules
- How You View WFM Team Schedules
- Process Imported Shifts Process

How You Set Up Shift Owners to Support Third-Party Schedule Imports

For each third-party scheduling application that you import shifts from, you need to add codes to the ORA_HWM_SHIFT_ENTRY_OWNER lookup. WFM schedules use these codes to identify the owner of each shift, so schedulers know the source of the shift data.

1. In the Setup and Maintenance work area, on the Tasks panel tab, click **Search**.
2. Search for and click **Manage Common Lookups**.
3. On the Manage Common Lookups page in the Lookup Type field, enter **ORA_HWM_SHIFT%**.
4. Click **Search**.

Related Topics

- How You Plan and Publish WFM Schedules
- Manager-Specific Options on WFM Team Schedules

How You Set Up Alert Notifications to Support Third-Party Schedule Imports

You can automatically notify schedulers about newly imported schedules. You can also automatically notify people when they have newly published schedules. Configure the recipients and messages for these alerts in the **Tools** > **Alerts Composer** page:

- HTS Worker Shifts Imported
- HTS Schedule Publication

Related Topics

- Overview of Importing Third-Party Schedules

How You Import Third-Party Schedules

You can import up to four levels of schedule business objects using HCM Data Loader and REST services. Both of these import methods include validation, such as expected data formats and values.

You can view any data loader errors using the Data Exchange work area. The REST APIs return error messages for data that fails validation.

HCM Data Loader

To upload full schedules that cover a long period for multiple people during your implementation, use the Import and Load Data task. You can also use this task to upload full schedules once a week, ongoing. For details on importing Schedule business objects to the workforce management server, see the HCM Data Loading Business Objects guide in *Oracle Help Center*.

To create the .dat files for HCM Data Loader, you can use HCM Spreadsheet Data Loader and the Schedule Request business object. For details on spreadsheet templates, see the HCM Data Loader guide in *Oracle Help Center*.

REST API Schedule Resources and Requests

To sync regular updates for one or several shifts at a time, you can set up REST Schedule resources and requests. For details on this configuration, see the REST API for Oracle Global Human Resources Cloud guide in *Oracle Help Center*.

Related Topics

- Overview of Importing Third-Party Schedules
- Process Imported Shifts Process

12 Loading Third-Party Time Cards

Overview of Loading Read-Only Time Cards Using HCM Data Loader

You can view and report on third-party time cards for historical purposes in Oracle Fusion Cloud Time and Labor.

Use the Time Record Group business object for the Workforce Management product area and HCM Data Loader. Also use HCM Data Loader to modify any third-party time cards you loaded this way.

Tip: To load third-party time cards for processing and payment, use the Time REST resources.

Calculated Time

Any applicable time calculation rules run for loaded time cards and generate calculated time. If no time calculation rules apply, calculated time is created by copying reported time.

Time Data Doesn't Transfer to Consumers

The data from third-party time cards loaded and modified with HCM data loader is for reference. It doesn't transfer to time consumers and doesn't get used to pay people or to bill customers.

Supported Time Card Periods

The load process supports time card periods that can get linked to time processing profiles, which are weekly, biweekly, and monthly. The Resubmit Time Card process can create single-day time cards when it splits an existing time card because of retroactive changes. These single-day time cards still fall within a weekly, biweekly, or monthly time card period. Single-day entries also need to be part of these supported time card periods.

Related Topics

- Example of Loading Read-Only Payroll Time Cards
- Example of Loading Read-Only Project Costing Time Cards

Example of Loading Read-Only Payroll Time Cards

This example TimeRecordGroup.dat file creates a Time card object with all of the applicable time entries. The file identifies both the Time card object and the linked time entries by their source keys.

```
METADATA|TimeRecordGroup|PersonNumber|ResourceType|TcStartTime|TcStopTime|GroupType|AssignmentNumber
MERGE|TimeRecordGroup|17772|PERSON|2017/12/04 08:00:00|2017/12/10 23:59:00|RPTD_TIME|E17772-2
METADATA|TimeRecord|PersonNumber|ResourceType|TcStartTime|TcStopTime|GroupType|OrderEntered|StartTime|
StopTime|Measure|UnitOfMeasure|AssignmentNumber|TmRecType
MERGE|TimeRecord|17772|PERSON|2017/12/04 08:00:00|2017/12/10 23:59:00|RPTD_TIME|1|2017/12/05 08:00:00|
2017/12/05 12:00:00|4|HR|E17772-2|RANGE
```

```
METADATA|TimeRepositoryAttribute|PersonNumber|ResourceType|TcStartTime|TcStopTime|GroupType|OrderEntered|
AttributeName|AttributeDataType|AttributeLongValue|AttributeStringValue|AttributeBigDecimalValue|
AttributeDateValue|AttributeTimestampValue
MERGE|TimeRepositoryAttribute|17772|PERSON|2017/12/04 08:00:00|2017/12/10 23:59:00|RPTD_TIME|1|
PayrollTimeType|String||ZOTL_Regular|||
MERGE|TimeRepositoryAttribute|17772|PERSON|2017/12/04 08:00:00|2017/12/10 23:59:00|RPTD_TIME|1|LDG_ID|
BIG_DECIMAL|||202||
```

Related Topics

- Overview of Loading Read-Only Time Cards Using HCM Data Loader

Example of Loading Read-Only Project Costing Time Cards

This example TimeRecordGroup.dat file creates a Time card object with all of the applicable time entries for third-party historical time cards. The file identifies both the Time card object and the linked time entries by their source keys.

```
METADATA|TimeRecordGroup|GroupType|ResourceType|PersonNumber|TcStartTime|TcStopTime|AssignmentNumber
MERGE|TimeRecordGroup|TIME_RPTD|PERSON|468|2016/12/19 00:00:01|2016/12/25 23:59:59|E300000026957492-3
METADATA|TimeRecord|PersonNumber|ResourceType|TcStartTime|TcStopTime|GroupType|StartTime|StopTime|TmRecType|
OrderEntered|UnitOfMeasure|Measure|AssignmentNumber
MERGE|TimeRecord|468|PERSON|2016/12/19 00:00:01|2016/12/25 23:59:59|TIME_RPTD|2016/12/20 09:00:01|2016/12/20
 18:00:00|MEASURE|1|HR|8|E300000026957492-3
METADATA|TimeRepositoryAttribute|ResourceType|TcStartTime|TcStopTime|OrderEntered|GroupType|AttributeName|
AttributeDataType|AttributeStringValue|AttributeLongValue|AttributeBigDecimalValue|AttributeDateValue|
AttributeTimestampValue|AttributeUsageId|TimeRecordId|PersonNumber
MERGE|TimeRepositoryAttribute|PERSON|2016/12/19 00:00:01|2016/12/25 23:59:59|1|TIME_RPTD|PJC_PROJECT_ID|
BIG_DECIMAL|||Attrib_Bg_Pjc_Id|||||468
MERGE|TimeRepositoryAttribute|PERSON|2016/12/19 00:00:01|2016/12/25 23:59:59|1|TIME_RPTD|PJC_TASK_ID|
BIG_DECIMAL|||Attrib_Bg_Pjc_Task_Id|||||468
MERGE|TimeRepositoryAttribute|PERSON|2016/12/19 00:00:01|2016/12/25 23:59:59|1|TIME_RPTD|PJC_PROJECT_UNIT|
BIG_DECIMAL|||Attrib_Bg_Pjc_Unit|||||468
MERGE|TimeRepositoryAttribute|PERSON|2016/12/19 00:00:01|2016/12/25 23:59:59|1|TIME_RPTD|
PJC_EXPENDITURE_TYPE_ID|BIG_DECIMAL|||Attrib_Pjc_Exp_Id|||||468
MERGE|TimeRepositoryAttribute|PERSON|2016/12/19 00:00:01|2016/12/25 23:59:59|1|TIME_RPTD|
PJC_EXPENDITURE_TYPE_NAME|STRING|Attrib_Pjc_Exp_Type_Name|||||||468
MERGE|TimeRepositoryAttribute|PERSON|2016/12/19 00:00:01|2016/12/25 23:59:59|1|TIME_RPTD|
PJC_SYSTEM_LINKAGE_FUNCTION|STRING|Attrib_Pjc_System_Linkage|||||||468
```

Related Topics

- Overview of Loading Read-Only Time Cards Using HCM Data Loader

13 Loading Event Groups and Actions for Time Card Resubmission

Guidelines for Loading Event Groups Related to Time Card Resubmission

Event groups correlate event points, such as changes to someone's assignment record, which would require a retroactive recalculation of time cards. Load these groups using HCM Data Loader and a .zip file that contains the EventGroups.dat file.

> **Note:** Load the .zip file, not the EventsGroup.dat file.

The EventGroup business object includes these attributes:

- **DateTrackedEvent**: Attribute to identify data changes that start retroactive time card calculations, associated with the EventGroup through the EventGroupCode
- **EventValueChange**: Criteria to qualify a change that starts an event
- **EventValueQualifier**: Other contextual information to enforce when deciding the validity of an event

You can omit EventValueChange and EventValueQualifier attributes in the METADATA and MERGE rows as appropriate. For example, When the FromValue and ToValue attribute values are both <ANY_VALUE>, you don't need to add the METADATA and MERGE rows for EventValueChange. The load process has a business rule that when EventValueChange doesn't have a row for a DateTrackedEvent, it implies an <ANY_VALUE> to <ANY_VALUE> change.

> **Tip:** To identify that the event group is for time card resubmission you need to make the EventGroupType attribute value **A**.

Event Group Attributes and Possible Values for Time Card Resubmissions

These are the possible values you can provide for EventGroup, DateTrackedEvent, EventValueChange, and EventValueQualifier objects in the context of time card resubmissions.

Event Group

Attribute Name	Possible Values
EVentGroupCode	Any value
EventGroupName	Any value

Attribute Name	Possible Values
	The value can be translated. Include translated data using the EventGroupTranslation attribute. Load the data using the EventGroupTranslation.dat file after you create the event groups.
EventGroupType	A Always use this value for time card resubmissions.

Date Tracked Event

Attribute Name	Possible Values
ColumnName	See *WFM Events Mapping to Columns and Entities*
UpdateType	DT_UPDATE_COLUMN DT_CORRECTION DT_INSERT When using DT_INSERT, leave ColumnName blank.
EventGroupCode	Use the same value as the EventGroupCode in the EventGroup object row. It's the link between the two business objects.
DatedObjectName	See *WFM Events Mapping to Columns and Entities*

Event Value Change

Attribute Name	Possible Values
ColumnName	See *WFM Events Mapping to Columns and Entities*
UpdateType	DT_UPDATE_COLUMN DT_CORRECTION DT_INSERT When using DT_INSERT, leave ColumnName blank.
EventGroupCode	Use the same value as the EventGroupCode in the EventGroup object row. It's the link between the two business objects.
DatedObjectName	See *WFM Events Mapping to Columns and Entities*
Sequence	Any integer Identifies when to evaluate the change

Attribute Name	Possible Values
ValidEvent	Y, N
FromValue	<ANY_VALUE> Or any valid value for the selected column
ToValue	<ANY_VALUE> Or any valid value for the selected column
ParentEvtValChangeSequence	Leave blank
EffectiveStartDate	Date with YYYY/MM/DD format, such as 1999/11/25
EffectiveEndDate	Date with YYYY/MM/DD format, such as 1999/11/25

Event Value Qualifier

Attribute Name	Possible Values
ColumnName	See *WFM Events Mapping to Columns and Entities*
UpdateType	DT_UPDATE_COLUMN DT_CORRECTION DT_INSERT When using DT_INSERT, leave ColumnName blank.
EventGroupCode	Use the same value as the EventGroupCode in the EventGroup object row. It's the link between the two business objects.
DatedObjectName	See *WFM Events Mapping to Columns and Entities*
Sequence	Any integer Identifies when to evaluate the change
EffectiveStartDate	Date with YYYY/MM/DD format, such as 1999/11/25
EffectiveEndDate	Date with YYYY/MM/DD format, such as 1999/11/25
QualifierName	Name of the provided qualifier, such as 'EmployeeAssignmentDEO Primary Flag'
QualifierValue	Y, N

Example of Loading an Event Group for Time Card Resubmission

This example EventGroups.dat file includes event value and qualifier attributes. Remember, to identify that the event group is for time card resubmission you need to make the EventGroupType attribute value A.

```
METADATA|EventGroup|EventGroupCode|EventGroupName|EventGroupType
MERGE|EventGroup|JobChangeEventGroup|JobChangeEventGroup|A

METADATA|DateTrackedEvent|ColumnName|UpdateType|EventGroupCode|DatedObjectName
MERGE|DateTrackedEvent|JOB_ID|DT_UPDATE_COLUMN|JobChangeEventGroup|
oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignmentDEO

METADATA|EventValueChange|ColumnName|UpdateType|EventGroupCode|DatedObjectName|Sequence|ValidEvent|
FromValue|ToValue|ParentEvtValChangeSequence|EffectiveStartDate|EffectiveEndDate
MERGE|EventValueChange|JOB_ID|DT_UPDATE_COLUMN|JobChangeEventGroup|
oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignmentDEO|1|Y|<ANY_VALUE>|<ANY_VALUE>||
1950/01/01|4712/12/31

METADATA|EventValueQualifier|ColumnName|UpdateType|EventGroupCode|DatedObjectName|Sequence|
EffectiveStartDate|EffectiveEndDate|QualifierName|QualifierValue
MERGE|EventValueQualifier|JOB_ID|DT_UPDATE_COLUMN|JobChangeEventGroup|
oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignmentDEO|1|1950/01/01|4712/12/31|
EmployeeAssignmentDEO Primary Flag|Y
```

WFM Events Mapping to Columns and Entities

This reference shows you the column name and dated object name for each WFM event that you can use for time card resubmissions.

WFM Event	ColumnName	DatedObjectName
Marital status	MARITAL_STATUS	oracle.apps.hcm.people.core.protectedModel.entity.PersonLegislativeInf
Home Location	ADDRESS_ID	oracle.apps.hcm.addresses.publicModel.entity.AddressDEO
Business Unit	BUSINESS_UNIT_ID	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Grade Ladder	GRADE_LADDER_PGM_ID	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Working Hours	NORMAL_HOURS	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Standard Working Hours	TOTAL_HOURS	oracle.apps.hcm.employment.core.publicModel.entity.WorkingHourPatte
Job	JOB_ID	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Grade	GRADE_ID	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Is Manager	MANAGER_FLAG	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Location	LOCATION_ID	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm

WFM Event	ColumnName	DatedObjectName
Position	POSITION_ID	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Worker Category	EMPLOYEE_CATEGORY	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Assignment Category	EMPLOYMENT_CATEGORY	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Regular or Temporary	PERMANENT_TEMPORARY_FLAG	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Full-Time or Part-Time	FULL_PART_TIME	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Person Type	PERSON_TYPE_ID	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Hourly Paid or Salaried	HOURLY_SALARIED_CODE	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Frequency	FREQUENCY	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Working at Home	WORK_AT_HOME	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Legal Employer	LEGAL_ENTITY_ID	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Department	ORGANIZATION_ID	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Reporting Establishment	ESTABLISHMENT_ID	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Bargaining Unit	BARGAINING_UNIT_CODE	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Collective Agreement	COLLECTIVE_AGREEMENT_ID	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Union Member	LABOR_UNION_MEMBER_FLAG	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Overtime Period	OVERTIME_PERIOD	oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignm
Manager	MANAGER_ID	oracle.apps.hcm.employment.core.publicModel.entity.AssignmentSuper
Work Day Definition	WORK_DAY_DEF_ID	oracle.apps.hcm.employment.core.publicModel.entity.WorkingHourPatte

Guidelines for Loading Event Actions Related to Time Card Resubmission

Event actions define how to react to an event, for example, whether to calculate retroactive payroll or recalculate time cards. Load these actions using HCM Data Loader and a .zip file that contains the EventActions.dat file.

Note: Load the .zip file, not the EventsActions.dat file.

The EventActions business object includes these attributes:

- **EventActionCriteria**: Specifies contextual conditions, such as starting retroactive calculations of time cards for members of specific HCM groups
- **EventGroupAction**: Association between an action and event group

You can omit EventActionCriteria attributes in the METADATA and MERGE rows as appropriate.

Event Action Attributes and Possible Values for Time Card Resubmissions

These are the possible values you can provide for EventAction, EventGroupAction, and EventActionCriteria objects in the context of time card resubmissions.

Event Group

Attribute Name	Possible Values
EventActionCode	Any value
EffectiveStartDate	Date with YYYY/MM/DD format, such as 1999/11/25
EffectiveEndDate	Date with YYYY/MM/DD format, such as 1999/11/25
EventActionName	Any value The value can be translated. Include translated data using the EventActionTranslation attribute. Load the data using the EventActionTranslation.dat file after you create the event actions.
EventActionTypeCode	ORA_HWM_ACTION_TYPE_TC_RESUBMIT Always use this value for time card resubmissions.
LookbackTimeDefinitionCode	12_MONTHS_AGO_MONTH_START_DATE 6_MONTHS_AGO_MONTH_START_DATE 3_MONTHS_AGO_MONTH_START_DATE This value indicates how far in the past from the action run date the process should look for time cards to resubmit. For example, a job change is effective on January 1, 2023. If the time card resubmission process runs on January 1, 2024, with LookbackTimeDefinitionCode set to 3_MONTHS_AGO_MONTH_START_DATE, the process considers only time cards from October 1, 2023 for resubmission.
ActionSubmission	SYNC Always use this value for time card resubmission.
ProcessMode	AUTO, MANUAL • AUTO: The process automatically resubmits the time card. • MANUAL: The process sets the time card Resubmission status to Resubmit, indicating that the time card needs to be manually resubmitted.

Attribute Name	Possible Values
AutoApprove	Y, N If Y, the process automatically set approved time cards back to the Approved status after resubmission.

Event Group Action

Attribute Name	Possible Values
EventActionCode	Use the same value as the EventActionCode in the EventAction object row. It's the link between the two business objects.
EffectiveStartDate	Date with YYYY/MM/DD format, such as 1999/11/25
EffectiveEndDate	Date with YYYY/MM/DD format, such as 1999/11/25
EventGroupCode	Use the same value as the EventGroupCode in the EventGroup object row. It's the link between the two business objects.

Event Action Criteria

Attribute Name	Possible Values
EventActionCode	Use the same value as the EventActionCode in the EventAction object row. It's the link between the two business objects.
EffectiveStartDate	Date with YYYY/MM/DD format, such as 1999/11/25
EffectiveEndDate	Date with YYYY/MM/DD format, such as 1999/11/25
SourceCode	Group code for the specified HCM GROUP. See the GRP_CODE column in the HWM_GRPS_VL table. The action will run only for workers belonging to the specified group.

Example of Loading an Event Action for Time Card Resubmission

This example EventActions.dat file includes event action criteria and group action attributes.

Tip: EventActionCriteria rows are optional, include them when appropriate.

```
METADATA|EventAction|EventActionCode|EffectiveStartDate|EffectiveEndDate|EventActionName|
EventActionTypeCode|LookbackTimeDefinitionCode|ActionSubmission|ProcessMode|AutoApprove
```

```
MERGE|EventAction|JobChangeAction|1950/01/01|4712/12/31|JobChangeAction|ORA_HWM_ACTION_TYPE_TC_RESUBMIT|
12_MONTHS_AGO_MONTH_START_DATE|SYNC|MANUAL|Y

METADATA|EventGroupAction|EventActionCode|EffectiveStartDate|EffectiveEndDate|EventGroupCode
MERGE|EventGroupAction|JobChangeAction|1950/01/01|4712/12/31|JobChangeEventGroup

METADATA|EventActionCriteria|EventActionCode|EffectiveStartDate|EffectiveEndDate|SourceCode
MERGE|EventActionCriteria|JobChangeAction|1950/01/01|4712/12/31|ABC_GRP
```

14 Loading Payroll Relationships

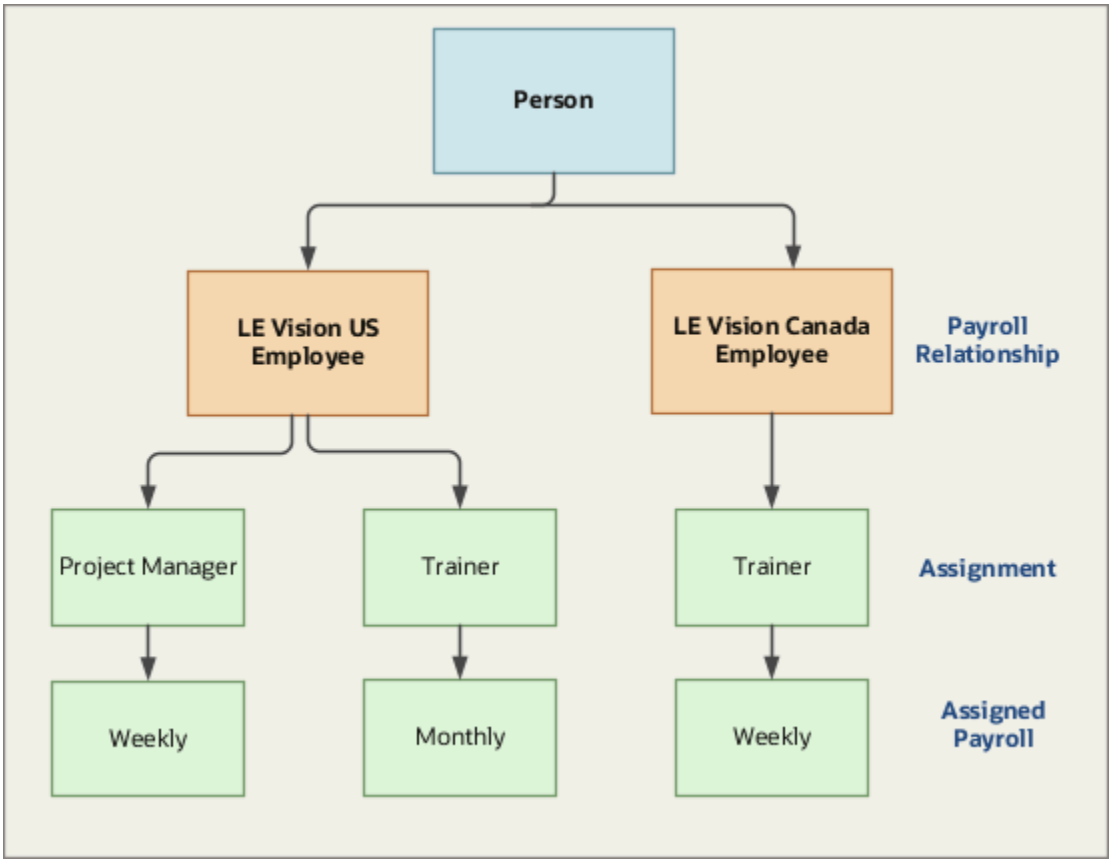
Payroll Information

Overview of Loading Payroll Details

Use HCM Data Loader to load payroll information at the payroll relationship, payroll assignment records, and assigned payroll levels. For example, you can update the overtime period for employees either at their payroll relationship or assignment level.

A time period you specify at a lower level, such as assignment level, overrides a period at a higher level, such as payroll relationship.

This figure shows the payroll information for a person at the Payroll Relationship, Assignment, and Assigned Payroll levels.



Consider these examples that explain the various actions you can take using HCM Data Loader.

- You manage Carrie Smith who's a part-time temporary employee and is on weekly payroll. Starting one month from now, Carrie accepted an offer to become a full-time permanent employee in the same position. You can update Carrie's assignment record and transfer her to a payroll appropriate for a full-time permanent employee, such as monthly or semimonthly.

- Anthony has two assignment records, one on a weekly payroll and one on a monthly payroll. On 10-Jun-18, you terminated Anthony's assignment record on the weekly payroll. The termination process automatically set the last standard process date to the end date of the payroll period. Anthony's termination package specifies that he should receive compensation payments through the month of June. To ensure that he's paid on both weekly and monthly payroll through June, you change the LSPD on weekly payroll to 30-Jun-18.
- You wrongly assigned Sophie, who's a new hire, to weekly payroll instead of monthly payroll. Now, you can delete her weekly payroll record and add the monthly payroll record.

Related Topics

- Example of Loading Element Duration Dates for Assigned Payroll
- Example of Loading Element Duration Dates For Payroll Assignments
- Overview of Implementing Payroll Relationship

Overview of Loading Payroll Relationship Details

Use HCM Data Loader to update payroll details for employees at the payroll relationship level.

Payroll relationships group a person's employment assignment records based on the payroll statutory calculation and reporting requirements. For example, you can update the overtime period for employees at their payroll relationship.

When an employee is hired, the application automatically creates the payroll relationship details. A person could have multiple payroll relationships if they're associated with more than one PSU or more than one payroll relationship type.

Element Duration Dates

The element duration dates at the payroll relationship level control the start and end date of earnings and deductions assigned to the employee's payroll relationship. Examples of payroll relationship elements include tax and social insurance entries.

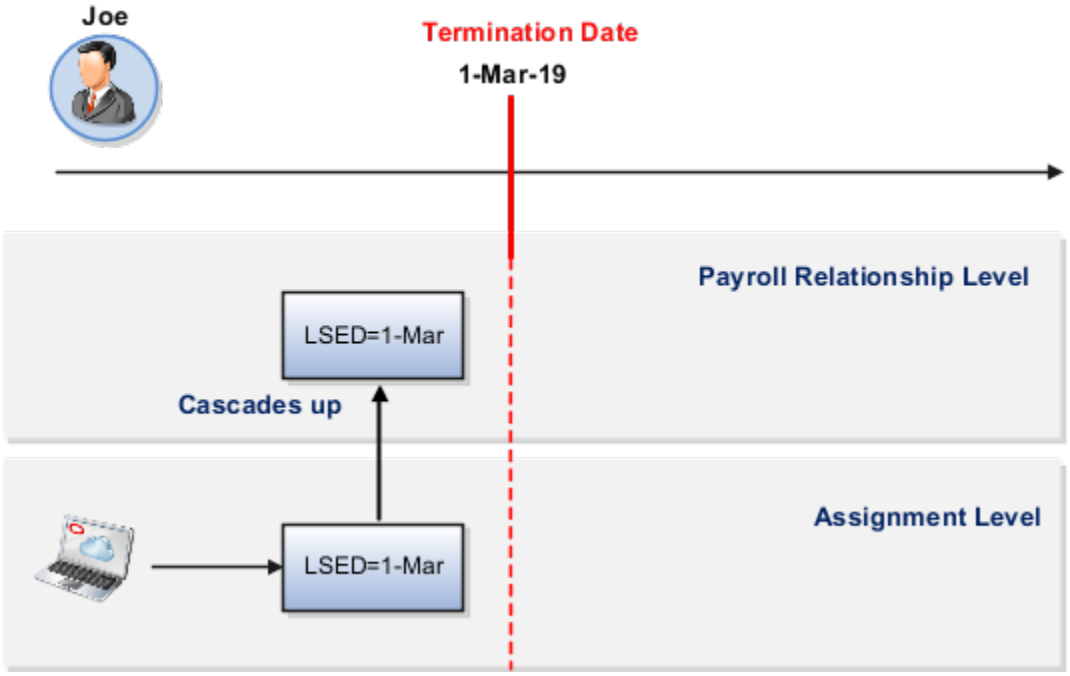
See this table for various element duration dates.

Date	Description
First Standard Earnings Date	The date when standard earnings start accumulating. For example, the start date of the employee's record for the payroll statutory unit.
Last Standard Earnings Date	The date when standard earnings stop accumulating. For example, the termination date of the employee's record for the payroll statutory unit.
Last Standard Process Date	The last date of the payroll period in which the termination of the payroll relationship occurs.
Final Close Date	The last date the payroll run will include elements (which are set to use final close) for processing. This includes regular and supplemental payroll runs

You can't edit these element duration dates at the payroll relationship level. These dates are populated based on the date information at the assignment level.

For example, when Joe is terminated on 1-Mar-19, the application sets the last standard earnings date on his assignment record with this date.

As the figure shows, the same date cascades up to the last standard earnings date field at the payroll relationship level.



Related Topics

- Overview of Loading Payroll Relationship Details
- Example of Loading Element Duration Dates for Assigned Payroll
- Example of Loading Element Duration Dates For Payroll Assignments
- Update the Time Card Required Option for Assignments

Assigned Payroll

Example of Loading Assigned Payroll Details

Use HCM Data Loader to assign a payroll to an assignment. Though you can specify the Time card Required and Overtime Status details at the assigned payroll level, consider defining them for an assignment.

For example, an employee is assigned to the weekly payroll and at the assigned payroll level, you load the overtime period code. If the employee transfers to the monthly payroll, the overtime period will no longer apply. In this case, you must define the overtime period at the assignment level.

Use HCM Data Loader to manage all payrolls for an assignment and for a given payroll relationship.

For example, in this DAT file, you assign a weekly payroll to an employee.

```
METADATA|AssignedPayroll|EffectiveStartDate|AssignmentNumber|PayrollDefinitionCode|LegislativeDataGroupName|
StartDate
```

```
MERGE|AssignedPayroll|2019/03/01|E10001|Weekly Payroll|Vision US LDG|2019/03/01
```

Example of Loading Element Duration Dates for Assigned Payroll

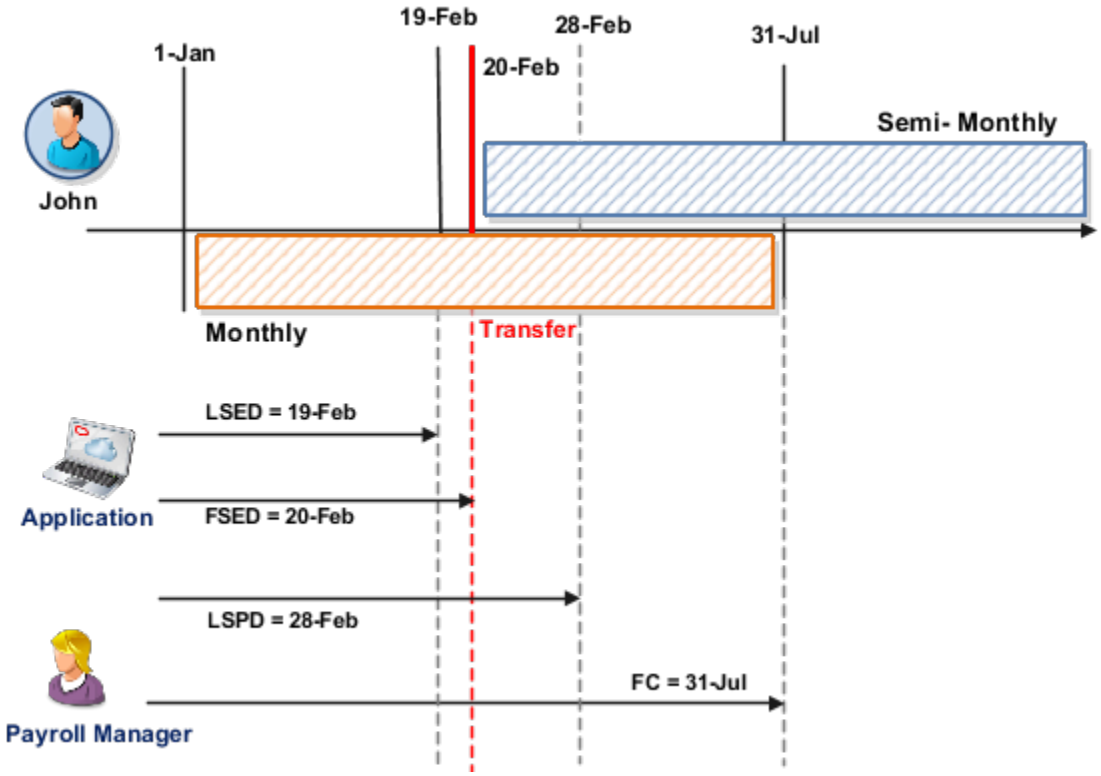
The element duration dates at assigned payroll level control when the assignment will be processed in a specific payroll run.

Let's consider this example.

John is on a monthly payroll, but starting 20-Feb-18, he's transferred to a semi-monthly payroll. His monthly and semi-monthly payroll definitions looks like this. You can update the final close date as 31-Jul-18.

Time Definition	Monthly Payroll	Semi-Monthly Payroll
First Standard Earnings Date	01/01/18	02/20/2018
Last Standard Earnings Date	02/19/2018	
Last Standard Process Date	02/28/2018	
Final Close	07/31/2018	

This image shows the various time definition dates for monthly and semi-monthly payroll.



Use this dat file to set the final close date as 31-Jul-18.

```
METADATA|AssignedPayroll|EffectiveStartDate|FinalCloseDate|LegislativeDataGroupName|PayrollDefinitionCode|
AssignmentNumber|StartDate
MERGE|AssignedPayroll|2018/01/01|2018/07/31 00:00:00|Vision US LDG|Monthly Payroll|E10003|2018/01/01
```

Payroll Assignments

Example of Loading Element Duration Dates For Payroll Assignments

Use HCM Data Loader to load payroll related information for an assignment, such as element duration dates. The element duration dates at assignment level control the start and end date of earnings and deductions assigned to the employee's assignment, such as salary.

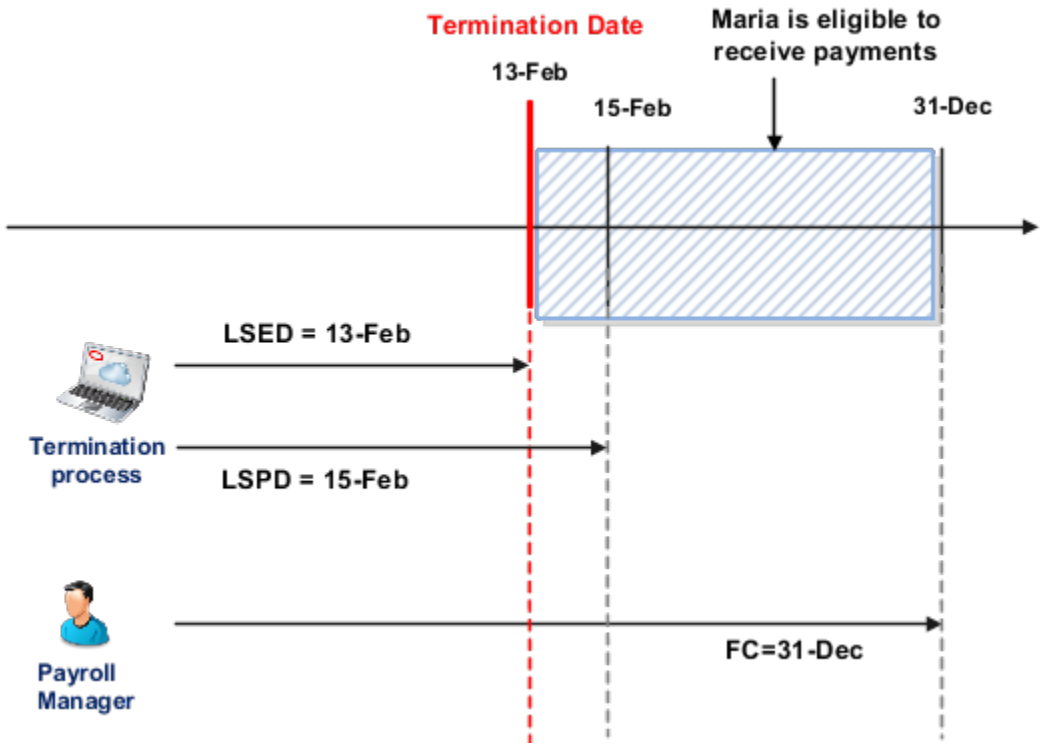
Example

On 13-Feb-2019, Maria Peterson who's on weekly payroll terminates her service with your company. The termination process automatically sets these process dates:

- The last standard earnings date to 13-Feb-19.
- The last standard process date to the end date of her weekly payroll, which is 15-Feb-19.

However, Maria is still eligible to receive outstanding overtime and commission payments, which can be paid up to 6 months in arrears. To ensure that she's paid these payments, you can update the final close date on the assignment record to 31-Dec-19.

As this image shows, the termination process sets the process dates for Maria.



In this dat file, you update the final close date to 31-Dec-2019.

```
METADATA|ElementDurationDate|DateValue|SourceType|AssignmentNumber|TimeDefinitionCode
MERGE|ElementDurationDate|2019/12/31|PA|E10001|Final Close
```

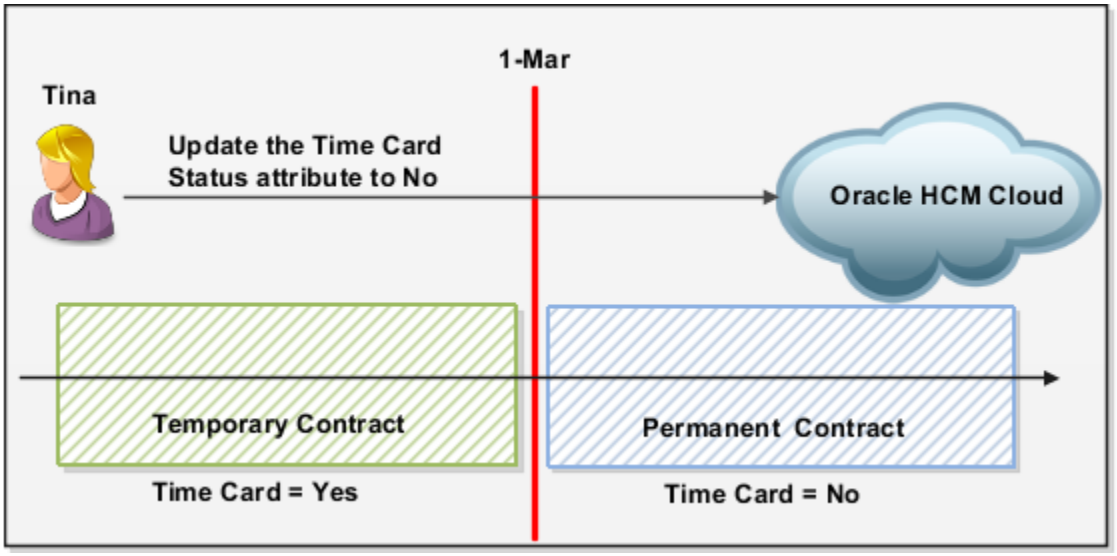
Related Topics

- Overview of Loading Payroll Relationship Details
- Update the Time Card Required Option for Assignments
- Update the Last Standard Process Date for Assignments

Update the Time Card Required Option for Assignments

You can use the **Time Card Required** option to indicate if the assignment is paid by time card or not. At the assignment level, the time card status applies to the specific assignment record.

Let's consider this example. Tina is on a temporary contract. She submits her time cards every week and receives her pay checks on a weekly basis. Effective 15-July, the company offers her a permanent contract. Under the new contract, she will become a salaried employee and will be paid on a monthly basis. And she will no longer need to submit the time card.



Use this dat file to set the **Time Card Required** parameter to **No**.

```
METADATA|PayrollAssignmentDetails|EffectiveStartDate|TimecardRequiredFlag|AssignmentNumber
MERGE|PayrollAssignmentDetails|2019/07/15|N|E10002
```

Related Topics

- Overview of Loading Payroll Relationship Details
- Update the Last Standard Process Date for Assignments

Update the Last Standard Process Date for Assignments

You can sequence the loading of the element duration dates depending on whether you move the element duration dates forward or backward.

This table describes the various actions involved in updating the last standard process date.

Action	What You Do	Example
Move the element duration dates forward.	Load the final close date first.	You update the final close date from 15-Mar to 15-May and the last standard process date from 15-Feb to 15-Apr.
Move the element duration dates backward.	Load the last standard process date first.	You update the last standard process date from 15-APR to 15-FEB and the final close date from 15-May to 15-Mar.

Let's consider this scenario where you hire an employee on 1-Jan-15 and create their Tax withholding calculation card. You terminate the employee on 15-Jan-16. Further, you update the final close date as 15-Mar-16 and the last standard process date as 15-Feb-16 at the assignment level. At a later point in time, you update the LSPD as 15-Apr-16 and FC as 15-May-16.

You must use these two DAT files to load the LSPD and FC date values separately.

```
METADATA|ElementDurationDate|DateValue|SourceType|AssignmentNumber|TimeDefinitionCode
MERGE|ElementDurationDate|2016/05/15|PA|E10001|Final Close
METADATA|ElementDurationDate|DateValue|SourceType|AssignmentNumber|TimeDefinitionCode
MERGE|ElementDurationDate|2016/04/15|PA|E10001|Last Standard Process Date
```

Related Topics

- Overview of Loading Payroll Relationship Details

15 Loading Payroll Setup Objects

Examples of Loading Legislative Data Groups

Legislative data groups are used to partition payroll and related data. At least one legislative data group is required for each country where the enterprise operates. Each legislative data group holds a legislation code, currency, and cost allocation key flexfield.

It's also associated with one or more payroll statutory units. This topic provides examples showing how to load and manage Legislative Data Group objects using HCM Data Loader.

Creating Legislative Data Groups

This example LegislativeDataGroup.dat file creates a legislative data group for the United Kingdom. It identifies the legislative data group using its source key. Codes are supplied for the currency and legislation.

```
METADATA|LegislativeDataGroup|SourceSystemOwner|SourceSystemId|Name|LegislationCode|DefaultCurrencyCode
MERGE|LegislativeDataGroup|VISION|LDG_VI_UK|Vision UK|GB|GBP
```

This example LegislativeDataGroup.dat file creates a legislative data group for the United States and associates it with a cost allocation key flexfield.

```
METADATA|LegislativeDataGroup|Name|Territory|DefaultCurrency|StructureInstanceName
MERGE|LegislativeDataGroup|Vision US|United States|US Dollar|System Test Costing
```

Deleting Legislative Data Groups

You can delete Legislative Data Group objects using HCM Data Loader. This example LegislativeDataGroup.dat file deletes a legislative data group. It identifies the legislative data group by its source key.

```
METADATA|LegislativeDataGroup|SourceSystemOwner|SourceSystemId
DELETE|LegislativeDataGroup|VISION|LDG_VI_UK
```

Object Groups

Overview

Use HCM Data Loader to define object groups that define subsets of people or elements for processing or reporting.

Define one of these object groups.

- Element Group
- Payroll Relationship Group

- Process Information Group

> **Note:** If you're loading an object group with a large number of inclusions or exclusions, you're recommended to use the individual Object Group Amend file to achieve that. This supports the data to be multithreaded and minimizes the processing time.

Element Group

An element group defines a group of individual elements for processing payroll runs or reports. Element groups are static, which means you select each element manually to include in a group.

Let's consider this example. In an off-cycle payroll run, you want to make expense payments that aren't subject to tax and other deductions. You create an element group for all expense elements. And you submit a payroll run with this element group as the parameter.

Use this .dat file to create an element group for all your required expenses.

```
METADATA|ObjectGroup|LegislativeDataGroupName|ObjectGroupCode|ObjectGroupName|ObjectGroupTypeCode|StartDate|
EndDate|StaticDynamicFlag|Description
MERGE|ObjectGroup|US LDG|VIS_EXP_GROUP|EXPENSE_ELEMENT_GRP|ELEGRP|2010/01/01||N|Expense elements group for
 off-cycle payments.
METADATA|ObjectGroupAmend|LegislativeDataGroupName|ObjectGroupCode|ObjectGroupLevelName|ObjectCode|
IncludeOrExclude
MERGE|ObjectGroupAmend|US LDG|VIS_EXP_GROUP|Element|Travel_Allowance|I
MERGE|ObjectGroupAmend|US LDG|VIS_EXP_GROUP|Element|Per_Diem|I
```

Payroll Relationship Group

Use a payroll relationship group to limit the persons that you want to process for payroll, data entry, and reporting. You can define your payroll relationship group to be either static or dynamic. In a dynamic group, you select a formula that determines the people to include in the group. You can use either the Payroll Relationship Number or Assignment Number or both to create the group.

In this example, Vision Corporation wants to pay bonus to a group of 25 employees in an off-cycle payroll run. Here, you create a payroll relationship group for these employees. When submitting the payroll run, you select the payroll relationship group as the parameter.

Use this .dat file to create a payroll relationship group.

```
METADATA|ObjectGroup|LegislativeDataGroupName|ObjectGroupCode|ObjectGroupName|ObjectGroupTypeCode|StartDate|
EndDate|StaticDynamicFlag|Description
MERGE|ObjectGroup|US LDG|VIS_BONUS0919|Vision Bonus Group|PAYEMP|2019/09/01|2019/10/31|N|Employees who will
 get paid the bonus in Sept. 2019.
METADATA|ObjectGroupAmend|LegislativeDataGroupName|ObjectGroupCode|ObjectGroupLevelName|ObjectCode|
IncludeOrExclude
MERGE|ObjectGroupAmend|US LDG|VIS_BONUS0919|Payroll Relationship|8186051|I
MERGE|ObjectGroupAmend|US LDG|VIS_BONUS0919|Payroll Relationship|8186965|I
MERGE|ObjectGroupAmend|US LDG|VIS_BONUS0919|Payroll Relationship|8186043|I
MERGE|ObjectGroupAmend|US LDG|VIS_BONUS0919|Payroll Relationship|8187231|I
MERGE|ObjectGroupAmend|US LDG|VIS_BONUS0919|Payroll Relationship|8181098|I
```

Related Topics

- Example of Loading Object Groups
- Object Group HCM Data Loader Files for Bank Reprocessing

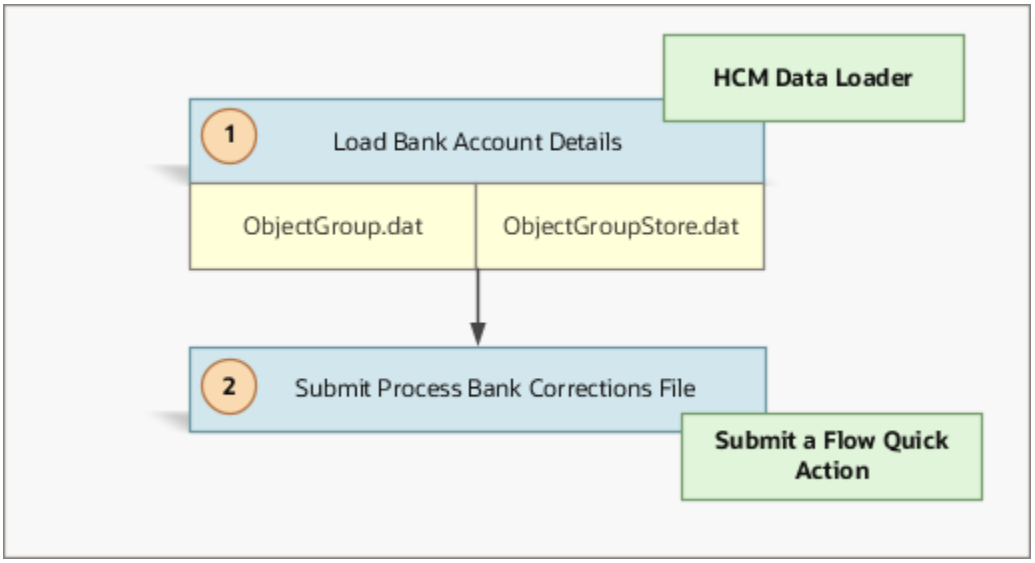
Example of Loading Object Groups

Rahul has an invalid branch number due to the closure of the bank.

And you want to update his bank account details. In this case, you don't end date the personal payment method, but update the existing personal payment method with the correct bank details.

To update Rahul's branch number, you do these tasks as shown in the figure.

1. Using HCM Data Loader, load the details returned by the bank.
2. Submit the **Process Bank Corrections File** flow to update his personal payment method.



Load the Bank Account Details

The bank returns a file containing details of the payments that are rejected. To process this bank information, use these .dat files.

1. ObjectGroup.dat file to create the object group.

```
METADATA|ObjectGroup|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|ObjectGroupCode|
ObjectGroupName|ObjectGroupTypeCode|Description|StartDate|EndDate|FormulaName
MERGE|ObjectGroup|VISION|PPM_E8185311|US LDG|PPM_E8185311|PPM Update for 8185311|PROCINFO|PPM update for
 employee 8185311|2019/10/01||
```

2. ObjectGroupStore.dat file to load the bank information returned by the bank

```
METADATA|ObjectGroupStore|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ObjectGroupId(SourceSystemId)|SequenceNumber|FLEX:PAY_OBJECT_GROUP_STORE_DDF|
amount(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)|
assignmentNumber(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)|
checkNumber(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)|
makeExternalPayment(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)|
paymentDate(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)|
paymentReference(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)|
processDate(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)|
replacementAccountType(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)|
replacementBranchNumber(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)|
returnreasoncode(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)|
```

```
employeeName(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)|
replacementAccountNumber(PAY_OBJECT_GROUP_STORE_DDF=ORA_PAY_BANK_CORRECTION)
MERGE|ObjectGroupStore|VISION|E8185311_BANK_CORRECTION|US LDG|PPM_E8185311|1|ORA_PAY_BANK_CORRECTION|
5000|E8185311|222324|YES|2019/10/31 00:00:00|185805|2019/10/31 00:00:00|Checking|004|AccountType Chg|
Rahul|00456754231
```

Submit the Process Bank Corrections File

On the Home page, click the **Submit a Flow** quick action under the **My Client Groups** tab. On the Flow Submission page, submit the **Process Bank Corrections File** flow. When you load the ObjectGroupStore.dat file into the Oracle Web Center Content server, a UCM ID is generated. The **Process Bank Corrections File** flow uses the UCM ID to retrieve and use the bank information necessary for the flow. The flow updates the Personal Payment Method with the new branch number details.

Related Topics

- Object Group HCM Data Loader Files for Bank Reprocessing

Example of Loading Payroll Consolidation Group

A payroll consolidation group is a grouping of payroll runs within the same time period for the same payroll. For this group, you can run reporting, costing, and post-run processing.

For example, a group might consolidate payroll runs for different work sites into one group for payments and costing.

Supply comments in a separate text file and then pass the text file name to the **Comments** attribute. Further, place the comments text file in the `ClobFiles` folder within the same compressed file as the PayrollConsolidationGroup.dat file that references it.

> **Note:** Don't supply the text directly in the PayrollConsolidationGroup.dat file.

Use this example dat file to load payroll consolidation groups:

```
METADATA|PayrollConsolidationGroup|SourceSystemOwner|SourceSystemId|LegislativeDataGroupId(SourceSystemId)|
ConsolidationSetName|ConsolidationSetCode|Comments
MERGE|PayrollConsolidationGroup|VISION|INFUSION_US_EMP|HDL_LDG_VISION_US|InFusion US Employee Group|InFusion
 US Employee Group|Consolidation_Group.txt
```

Related Topics

- Examples of Loading Legislative Data Groups
- How Pay Frequency Components Work Together
- Consolidation Groups for the US

Payroll Definitions and Time Periods

Payroll Definitions and Time Periods: Overview

Payroll definitions contain calendar and offset information that determine when to calculate and cost payments. It includes definition of payment frequency, processing schedule, and other parameters for a particular payroll.

For example, to pay employees semimonthly, you can create a payroll definition using the semimonthly payroll period type, ensuring that tax and other calculations produce correct results for those employees.

You can't change either the first effective start date or last effective end date for an existing payroll definition. Create payroll definition objects with effective start dates on or before the start dates of other objects that refer to your definitions.

Payroll Offset

Payroll offsets help you define your payroll cycle schedule. You can select for your payroll cycle events to occur on specific dates or be based on offsets from period start or end dates. Based on your choices and the number of calendar years you specify, the complete calendar for your payroll's lifetime is generated.

This table describes the predefined payroll cycle events that you can offset.

Date	Description
Planned Submission Date	The date on which the user will submit the payroll run.
Cutoff Date	The final date that payroll information can be entered for the payroll period.
Payslip Availability Date	The date on which payees can view payslips.
Payroll Run Date	The date the payroll calculation process uses to retrieve effective values, such as employee details.
Date Earned	The date on which the payroll process processes element entries for the payroll run. The date earned must be within the effective dates of the payroll period.
Date Paid	The date the employee is marked as paid. For check payments, this is the date that the check is valid for cash or deposit. For electronic funds transfer (EFT) payments, it's the transfer date.

Valid Payment Methods

You can specify an effective date that determines when your organization payment method will be used for this payroll definition. This date range must fit within both the effective date range of the payroll definition and the effective date range of the organization payment method.

Payroll Time Periods

Payroll Time periods represent the various dates within each payroll cycle and their frequency. Payroll time periods hold the payroll definition calendar that's based on the frequency of the payroll, such as weekly and monthly. For example, you can adjust payroll days to account for a bank holiday.

Related Topics

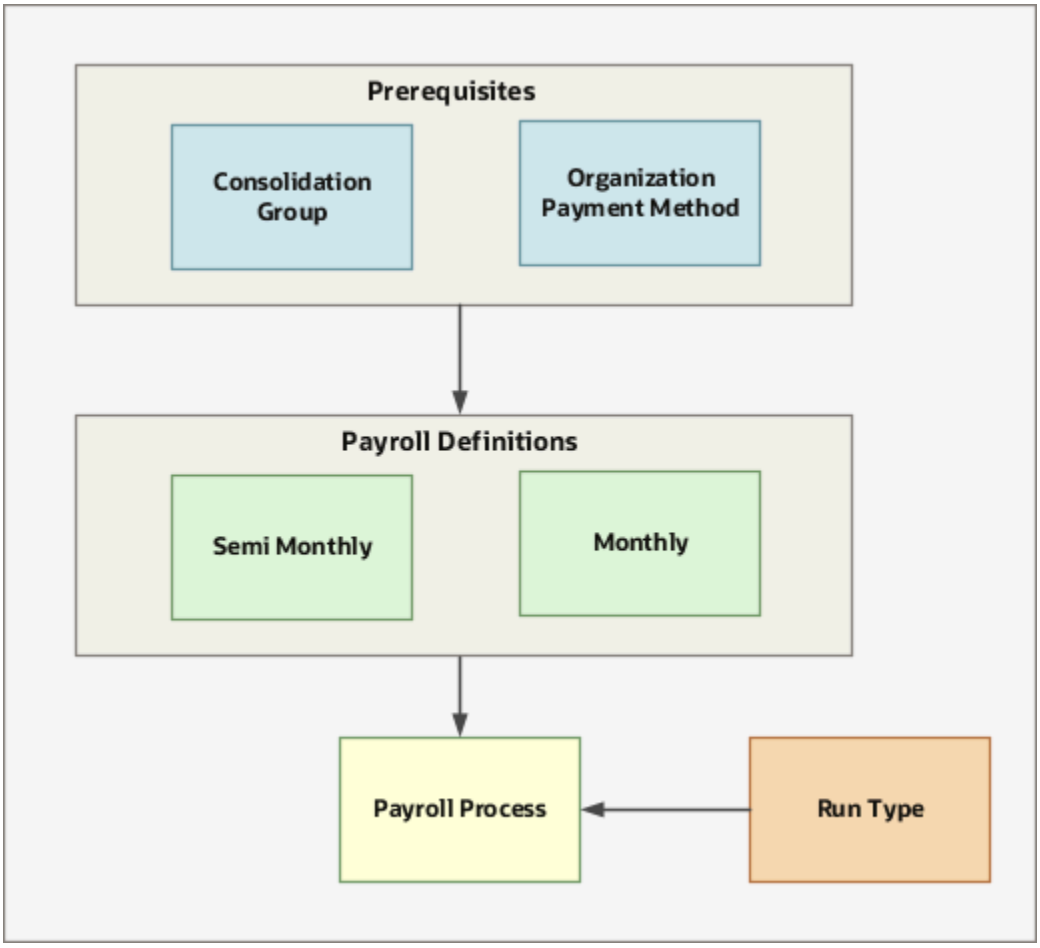
- Example of Loading Payroll Consolidation Group
- Payroll Definitions
- Overview of Payroll Definitions
- Create Payroll Definitions

Examples of Loading Payroll Definitions and Time Periods

In this example, the Vision Corp Company requires payroll definitions for two sets of employees.

One set is permanent salaried employees who are paid on a semimonthly basis, and the other is temporary employees that are paid on a monthly basis using time card data.

Note: The Company hires all employees after the effective start date of this payroll definition, so there is no issue with loading historical employee data.



Before using the HCM Data Loader to create payroll definitions, ensure that you load the following objects:

- Organization Payment Methods for your payrolls. For more information about loading default payment methods, refer to the topic: Loading Organization Payment Methods: Explained.
- Consolidation group that processes the results of one or more payroll runs in a single action. For more information about loading consolidation groups, refer to the topic: Loading Payroll Consolidation Group: Explained.

This example does these actions:

- Loads the payroll definition that pays employees into their bank accounts on a semimonthly basis.
- Includes dynamically generated offset dates.
- Identifies the default payment method for those employees who haven't chosen a personal payment method.
- Adjusts payroll days.

```
METADATA|PayrollDefinition|SourceSystemOwner|SourceSystemId|EffectiveStartDate|EffectiveEndDate|
PayrollCode|LegislativeDataGroupId(SourceSystemId)|ReportingName|ConsolidationGroupCode|PeriodTypeCode|
FirstPeriodEndDate|DefaultPaymentMethodId(SourceSystemId)|NumberOfYears|CutoffFalls|CutoffDayType|
CutoffOffset|CutoffBaseDate|PlannedSubmissionFalls|PlannedSubmissionDayType|PlannedSubmissionOffset|
PlannedSubmissionBaseDate|PayrollRunFalls|PayrollRunDayType|PayrollRunOffset|PayrollRunBaseDate|
PayslipAvailabilityFalls|PayslipAvailabilityDayType|PayslipAvailabilityOffset|PayslipAvailabilityBaseDate|
DateEarnedFalls|DateEarnedDayType|DateEarnedOffset|DateEarnedBaseDate|DatePaidFalls|DatePaidDayType|
DatePaidOffset|DatePaidBaseDate
```

```
MERGE|PayrollDefinition|VISION|HDL_INFUSION_USEMP_SEMIMONTH|2011/01/01|4712/12/31|
HDL_INFUSION_US_EMP_SEMIMONTHLY|HDL_LDG_VISION_US|InFusion US Employee Semimonthly|InFusion US Employee
 Group|Semi-Month|2012/06/15|HDL_INFUSION_US_EMP_EFT|5|5|W|B|E|4|W|B|E|0|W|B|E|0|W|B|E|0|W|B|E
```

Here, you specify the number of years ahead that the payroll cycle dates will be generated as 5. Also, you load the following offsets to your payroll processing schedule. In this example, the payroll cutoff date falls 5 days before the date earned for the payroll period.

Parameter	Falls Value	Day Type Value	Offset Value	Base Data Value
Cutoff Date	5	W	B	E
Planned Submission Date	4	W	B	E
Payroll Run Date	0	W	B	E
Payslip Availability Date	0	W	B	E
Date Earned	0	W	B	E
Date Paid	0	W	B	E

Load the payroll definition that pays employees by check and by using time card data on a monthly calendar basis.

```
METADATA|PayrollDefinition|SourceSystemOwner|SourceSystemId|EffectiveStartDate|EffectiveEndDate|
PayrollCode|LegislativeDataGroupId(SourceSystemId)|ReportingName|ConsolidationGroupCode|
PeriodTypeCode|FirstPeriodEndDate|DefaultPaymentMethodId(SourceSystemId)|NumberOfYears|FixedDate|
CutoffFixedDate|DateEarnedFixedDate|PayrollRunFixedDate|DatePaidFixedDate|PayslipAvailabilityFixedDate|
PlannedSubmissionFixedDate
MERGE|PayrollDefinition|VISION|HDL_INFUSION_USEMP_MONTH|2011/01/01|4712/12/31|HDL_INFUSION_US_EMP_MONTHLY|
HDL_LDG_VISION_US|InFusion US Employee Monthly|InFusion US Employee Group|Calendar Month|2012/06/30|
HDL_INFUSION_US_EMP_CHECK|5|Y|2012/06/25|2012/06/28|2012/06/28|2012/06/28|2012/06/28|2012/06/28 00:00:00
```

These payroll definition lines load fixed dates to define offsets:

Parameter	Value
Number of years	5
Set Date	Yes
Cutoff Date	2012/06/25
Date Earned	2012/06/28
Payroll Run Date	2012/06/28
Date Paid	2012/06/28

Parameter	Value
Payslip Availability Date	2012/06/28
Planned Submission Date	2012/06/28

Payroll Time Periods

As payroll time periods are generated when the payroll is defined, reference the payroll and time period to adjust using user keys. This is because the source key is generated by Oracle HCM cloud. The following DAT files explain how to adjust a time period for these payroll definitions.

```
METADATA|PayrollTimePeriod|PayrollCode|LegislativeDataGroupName|PeriodCategory|StartDate|PayrollRunDate|
DatePaid
MERGE|PayrollTimePeriod|HDL_INFUSION_US_EMP_SEMIMONTHLY|HDL Vision Clothing Corporation - US|E|2013/11/16|
2013/11/27|2013/11/27
MERGE|PayrollTimePeriod|HDL_INFUSION_US_EMP_MONTHLY|HDL Vision Clothing Corporation - US|E|2013/11/01|
2013/11/27|2013/11/27
```

Related Topics

- Payroll Definitions and Time Periods: Overview

Time Definitions

Example of Loading Time Definitions

Use HCM Data Loader to create time definitions that can be either a date or a span of time. You can define static definitions using multiple periods, additional adjustments, or both. Alternatively, you can also define dates based on dynamic variables.

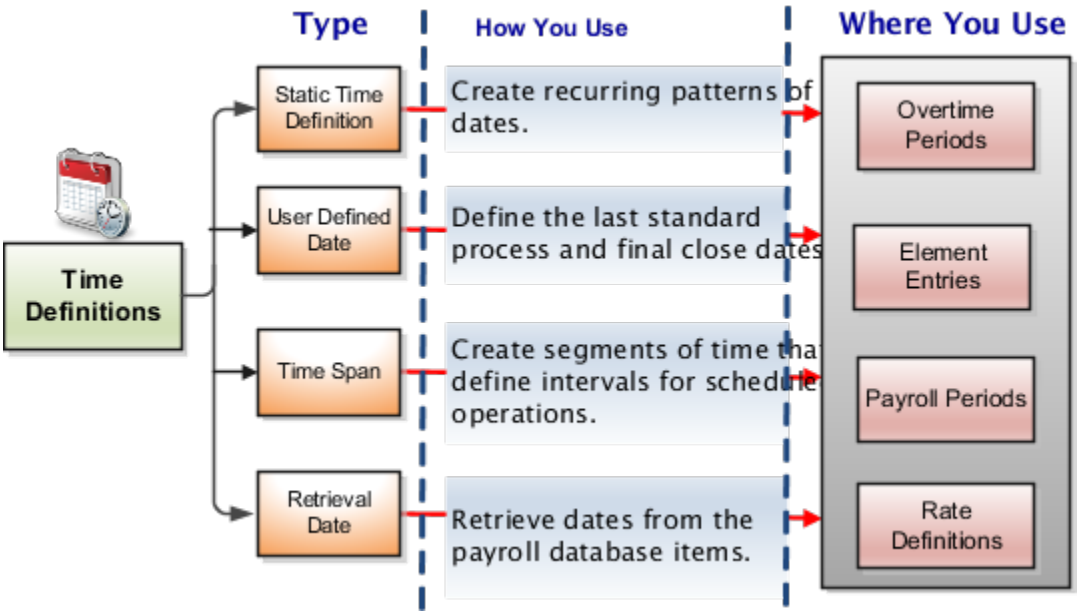
For example, you can create multiple time definitions to establish different workweeks for different facilities or group of employees.

Later, you use these time definitions in payroll periods, balances, overtime periods, balance dimensions, and so on.

You can load one of these time definitions:

- Static Time Definition
- User-Defined Date
- Time Span
- Retrieval Date

This image shows the different types of time definition and how and where you can use them.



Related Topics

- Example of Loading Static Time Definition
- Example of Loading User-Defined Date Time Definition
- Example of Loading Time Span Time Definitions
- Example of Loading Retrieval Date Time Definitions
- Use Time Definitions for Severance Pay

Example of Loading Static Time Definition

Load static time definitions to create recurring patterns of dates, such as overtime periods. These definitions can be static periods of unusual length based on a given date or they can generate dates based on dynamic variables.

A company's overtime period is a standard five day work week. You create a static time definition with a weekly frequency that generates time periods for 3 years.

```
METADATA|PayrollTimeDefinition|TimeDefinitionCode|DefinitionName|StartDate|DayAdjustment|DefinitionType|
NumberOfYears|PeriodType|PeriodUnit|BaseTimeDefinitionCode|DatabaseItemCode|PeriodTimeDefinitionCode|
AdjustmentType|LegislativeDataGroupName MERGE|PayrollTimeDefinition|RPGP_HDL_TD_002|RPGP_HDL_TD_002|
2016/01/01||Static time definition|2|Monthly|||||ABC US LDG
```

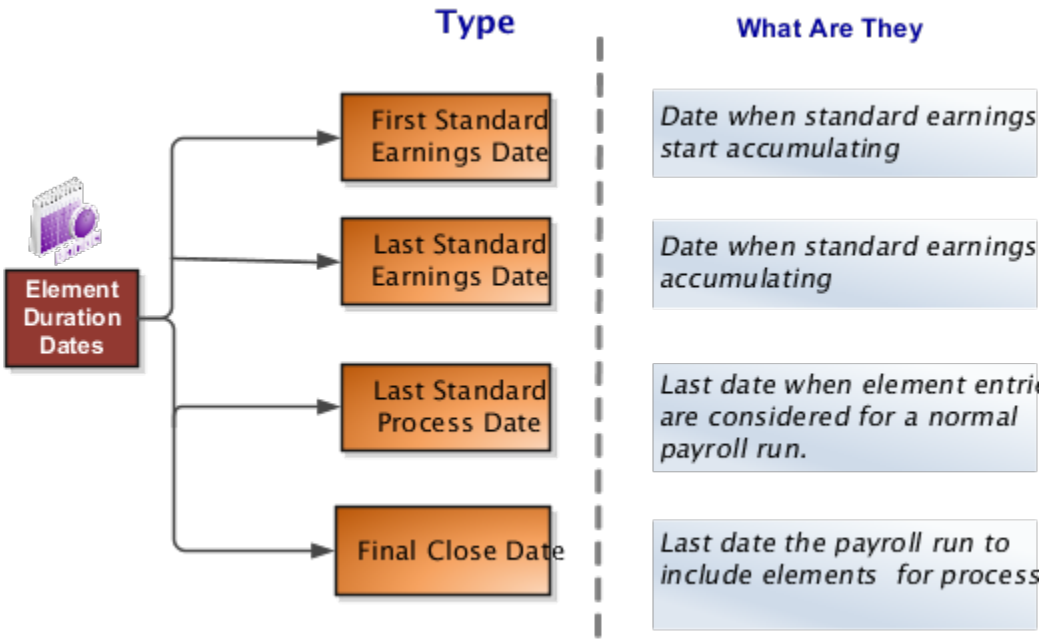
Related Topics

- Example of Loading User-Defined Date Time Definition
- Example of Loading Time Span Time Definitions
- Example of Loading Retrieval Date Time Definitions

Example of Loading User-Defined Date Time Definition

You can use these predefined element duration dates to control when element entries for an employee start or end.

This figure describes the various types of element duration dates.



This table provides examples on when you would use the predefined element duration dates.

Element Duration Date Type	Example
First Standard Earnings Date	Start date of the employee's record for the payroll statutory unit.
Last Standard Earnings Date	Termination date of the employee's record for the payroll statutory unit.
Last Standard Process Date	Last day of the payroll in which termination or payroll transfer occur.
Final Close Date	If a final close date is entered for an assigned payroll record, the employee will no longer be considered for processing by that specific payroll.

Using HCM Data Loader, you can load your own time definitions that define the last standard process and final close dates. For example, Vision Corp provides housing allowance to its employees and wants the allowance to end 30 days after the employee is terminated. You define a user-defined date time definition based on the last standard earnings date, which is the termination date plus 30 days.

```
METADATA|PayrollTimeDefinition|TimeDefinitionCode|DefinitionName|StartDate|DayAdjustment|DefinitionType|
NumberOfYears|PeriodType|PeriodUnit|BaseTimeDefinitionCode|DatabaseItemCode|PeriodTimeDefinitionCode|
AdjustmentType|LegislativeDataGroupName
```

```
MERGE|PayrollTimeDefinition|RPGP_HDL_TD_006|RPGP_HDL_TD_006||-2|User-defined date||User Year||First Standard
 Earning Date|||Subtract Days|ABC US LDG
```

Related Topics

- Example of Loading Time Span Time Definitions
- Example of Loading Retrieval Date Time Definitions
- Example of Loading Static Time Definition

Example of Loading Time Span Time Definitions

Use HCM Data Loader to create segments of time that define intervals for scheduled operations, such as payroll frequency. You can use these time span definitions within other time definitions.

A company wants to make employee payments on the last day of the month and third-party payments five days later. The company creates a time definition of type time span and uses it when defining the payment method for the payee. Use this `PayrollTimeDefinition .dat` file to create a time span time definition.

```
METADATA|PayrollTimeDefinition|TimeDefinitionCode|DefinitionName|StartDate|DayAdjustment|DefinitionType|
NumberOfYears|PeriodType|PeriodUnit|BaseTimeDefinitionCode|DatabaseItemCode|PeriodTimeDefinitionCode|
AdjustmentType|LegislativeDataGroupName
MERGE|PayrollTimeDefinition|RPGP_HDL_TD_005|RPGP_HDL_TD_005||11|Time span||Weekly|2||||Add Days|ABC US LDG
```

Related Topics

- Example of Loading Retrieval Date Time Definitions
- Example of Loading Static Time Definition
- Example of Loading User-Defined Date Time Definition

Example of Loading Retrieval Date Time Definitions

Load retrieval date time definition that's based on the payroll database items of date format.

Let's assume that the rate contributor for the severance rate definition uses a balance as of termination date. You can define a retrieval date time definition based on a database item that retrieves the termination date. You can then use this time definition as the reference date for the balance value in the rate contributor.

Use this .dat file to create a retrieval date time definition.

```
METADATA|PayrollTimeDefinition|TimeDefinitionCode|DefinitionName|StartDate|DayAdjustment|DefinitionType|
NumberOfYears|PeriodType|PeriodUnit|BaseTimeDefinitionCode|DatabaseItemCode|PeriodTimeDefinitionCode|
AdjustmentType|LegislativeDataGroupName
MERGE|PayrollTimeDefinition|RPGP_HDL_TD_001|RPGP_HDL_TD_001|||Retrieval Date|||||ENTRY_USAGE_END_DATE|||ABC
 US LDG
```

Example of Loading Payroll Element Run Usage

Use the payroll element run usage object to identify how an element is used with the run type. Run types control the elements and payment types to process in a payroll run.

You can't change either the first effective start date or last effective end date for an existing payroll element run usage object. Create these objects with effective start dates on or before the start dates of other objects that refer to your payroll element run usage.

Vision Corp processes and pays the bonus amounts separately from the regular earnings. The organization pays regular payroll by EFT and issues check bonuses once a year. For this requirement, they create a separate payment for the bonus element entry, which is marked to pay separately. Using HCM Data Loader, the Payroll Manager creates a bonus element, Annual Bonus, with Supplemental Earnings classification. She supplies these details.

Element Detail	Value
Primary Classification	Supplemental Earnings
Secondary Classification	Bonus
Category	Standard
Process and pay element separately or with other earnings elements?	Process separately and pay separately

Further, she loads the run type usage as **Separate Payment** with Element Usage as **Trigger**. The application processes this element as well as makes payments separately.

This example dat file loads the payroll element run usage object:

```
METADATA|PayrollElementRunTypeUsage|EffectiveStartDate|UsageType|ElementTypeCode|RunTypeCode|
LegislativeDataGroupName
MERGE|PayrollElementRunTypeUsage|2018/02/02|Trigger| ANNUAL_BONUS|Regular|Vision Corporation US
```

Related Topics

- Payroll Elements
- Run Types
- How Pay Frequency Components Work Together

User-Defined Tables

Overview

You can set up your own structured tables to maintain a date effective lists of values or data, such as wage codes. A user-defined table consists of rows and columns and stores the values as cells for a specific row and column combination.

For example, you can define a table that shows the bonus amounts in the columns, for the years of services in the rows.

Using HCM Data Loader, you can either create a user-defined table to store a range of values or match a specific value. You can use this table with a formula to store tabular data.

Rows, Columns, and Values

To start with, you define the column and row details. Let's understand what user-defined rows and columns are.

- A table row is a definition that stores row details. For example, a row that stores the years of service details for bonus payments. Define rows to accept either an exact value, such as a grade value, or a range of values, such as a range of salary values.
- A table column stores the column details, such as a column could capture bonus amounts. You can also use a fast formula to validate entries in the user-defined table. For example, you can validate if your bonus entries fall within a specified range.

Then, you load the values for a date-effective combination of a user-defined row and column.

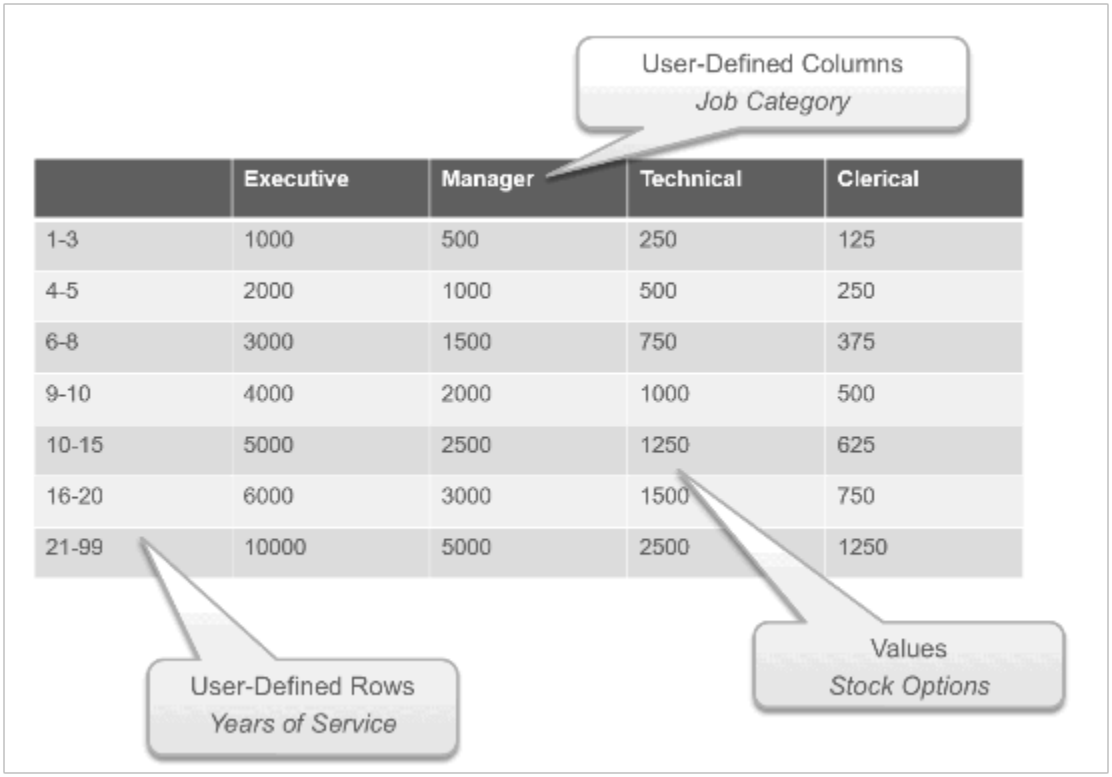
Related Topics

- Example of Loading User-Defined Tables for Matched Row Values
- Example of Loading User-Defined Tables for a Range of Row Values
- User Table Validation Formula Type

Example of Loading User-Defined Tables for a Range of Row Values

Let's consider this example. Each year, your organization offers stock options to its employees. The amount of stocks depend on the years of service and the job category of the employee receiving them.

As this image shows, the user-defined table contains stock option allocations by job category and years of service. The rows represent a range of years of service during which employees receive the same number of stock options. The columns contain the job categories. And the values represent the number of stock options awarded to the specified job category during the specified years of service.



Use this .dat file to load your stock option allocations.

```
METADATA|UserDefinedTable|LegislativeDataGroupName|UserTableCode|RangeOrMatch|UserKeyUnits|UserRowTitle|
UserTableName
MERGE|UserDefinedTable|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|R|N|Years of Service|
Employee Stock Options by Job Category and Length of Service

METADATA|UserDefinedTableColumn|LegislativeDataGroupName|UserTableCode|UserColumnCode|UserColumnName|
DataType|FormulaName
MERGE|UserDefinedTableColumn|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|EXECUTIVE|Executive|
N|
MERGE|UserDefinedTableColumn|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|MANAGER|Manager|N|
MERGE|UserDefinedTableColumn|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|TECHNICAL|Technical|
N|
MERGE|UserDefinedTableColumn|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|CLERICAL|Clerical|N|

METADATA|UserDefinedTableRow|LegislativeDataGroupName|UserTableCode|DisplaySequence|RowName|
EffectiveStartDate|EffectiveEndDate|RowLowRangeOrName|RowHighRange
MERGE|UserDefinedTableRow|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|10|1|2019/01/01|
4712/12/31|1|3
MERGE|UserDefinedTableRow|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|20|4|2019/01/01|
4712/12/31|4|5
MERGE|UserDefinedTableRow|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|30|6|2019/01/01|
4712/12/31|6|8
MERGE|UserDefinedTableRow|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|40|9|2019/01/01|
4712/12/31|9|10
MERGE|UserDefinedTableRow|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|50|11|2019/01/01|
4712/12/31|11|15
MERGE|UserDefinedTableRow|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|60|16|2019/01/01|
4712/12/31|16|20
MERGE|UserDefinedTableRow|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|70|21|2019/01/01|
4712/12/31|21|69

METADATA|UserDefinedTableColumnInstance|LegislativeDataGroupName|UserTableCode|RowLowRangeOrName|
UserColumnCode|EffectiveStartDate|EffectiveEndDate|Value
```

```
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|1|EXECUTIVE|
2019/01/01|4712/12/31|1000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|4|EXECUTIVE|
2019/01/01|4712/12/31|2000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|6|EXECUTIVE|
2019/01/01|4712/12/31|3000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|9|EXECUTIVE|
2019/01/01|4712/12/31|4000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|11|
EXECUTIVE|2019/01/01|4712/12/31|5000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|16|
EXECUTIVE|2019/01/01|4712/12/31|6000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|21|
EXECUTIVE|2019/01/01|4712/12/31|10000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|1|MANAGER|
2019/01/01|4712/12/31|500
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|4|MANAGER|
2019/01/01|4712/12/31|1000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|6|MANAGER|
2019/01/01|4712/12/31|1500
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|9|MANAGER|
2019/01/01|4712/12/31|2000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|11|MANAGER|
2019/01/01|4712/12/31|2500
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|16|MANAGER|
2019/01/01|4712/12/31|3000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|21|MANAGER|
2019/01/01|4712/12/31|5000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|1|TECHNICAL|
2019/01/01|4712/12/31|250
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|4|TECHNICAL|
2019/01/01|4712/12/31|500
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|6|TECHNICAL|
2019/01/01|4712/12/31|750
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|9|TECHNICAL|
2019/01/01|4712/12/31|1000
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|11|
TECHNICAL|2019/01/01|4712/12/31|1250
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|16|
TECHNICAL|2019/01/01|4712/12/31|1500
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|21|
TECHNICAL|2019/01/01|4712/12/31|2500
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|1|CLERICAL|
2019/01/01|4712/12/31|125
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|4|CLERICAL|
2019/01/01|4712/12/31|250
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|6|CLERICAL|
2019/01/01|4712/12/31|375
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|9|CLERICAL|
2019/01/01|4712/12/31|500
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|11|CLERICAL|
2019/01/01|4712/12/31|625
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|16|CLERICAL|
2019/01/01|4712/12/31|750
MERGE|UserDefinedTableColumnInstance|US Legislative Data Group|EMP_STOCK_OPT_BY_JOB_CAT_AND_LOS|21|CLERICAL|
2019/01/01|4712/12/31|1250
```

Related Topics

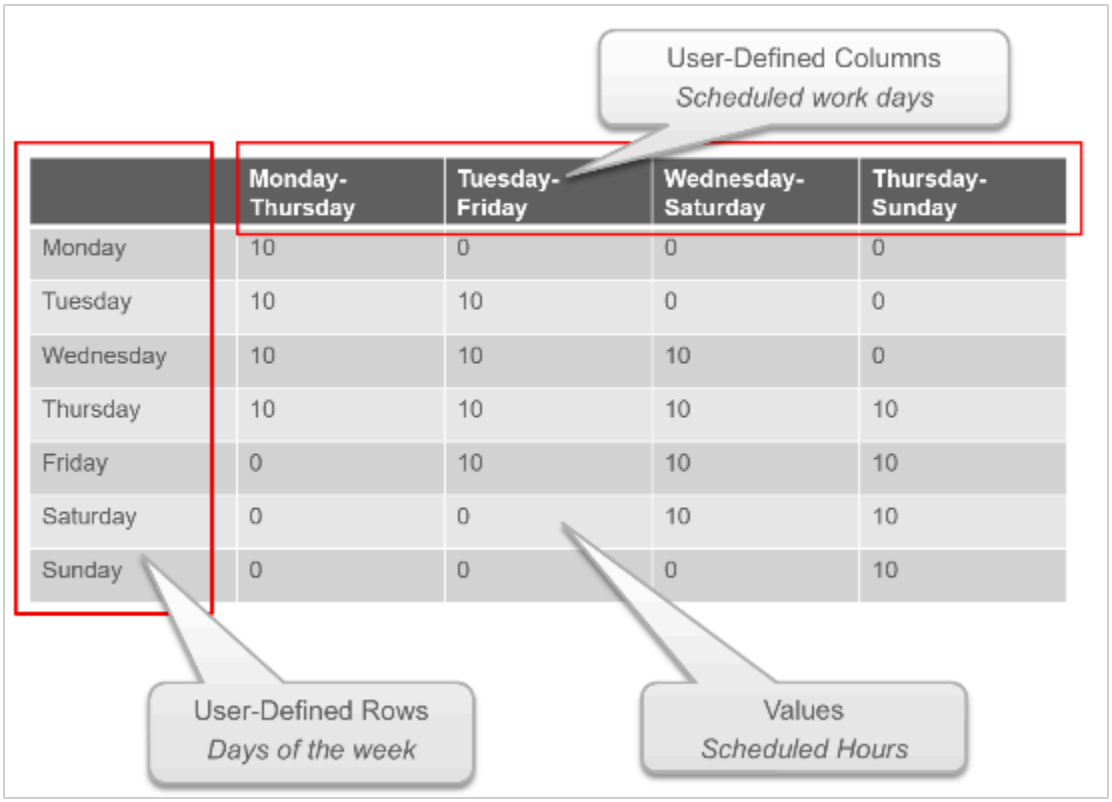
- Example of Loading User-Defined Tables for Matched Row Values
- User Table Validation Formula Type

Example of Loading User-Defined Tables for Matched Row Values

Vision Corporation wants to use a user-defined table to define schedules for their employees. All workers in the company work on a 10 hour a day, four day in a week rotating schedule. The company has these four schedules.

- Monday - Thursday
- Tuesday - Friday
- Wednesday - Saturday
- Thursday - Sunday

As this image shows, this user-defined table contains the schedules available in the organization. The rows contain the name of a day of the week and the columns contain the schedule. The values represent the number of hours to work each day in a schedule.



Use this .dat file to create a user-defined table to store values for worker's schedules.

```
METADATA|UserDefinedTable|UserTableCode|RangeOrMatch|UserKeyUnits|UserRowTitle|UserTableName|
LegislativeDataGroupName
METADATA|UserDefinedTableColumn|UserColumnCode|DataType|UserColumnName|UserTableCode|
LegislativeDataGroupName|FormulaName
METADATA|UserDefinedTableRow|DisplaySequence|EffectiveEndDate|EffectiveStartDate|RowHighRange|
RowLowRangeOrName|UserTableCode|LegislativeDataGroupName|RowName
METADATA|UserDefinedTableColumnInstance|EffectiveEndDate|EffectiveStartDate|Value|LegislativeDataGroupName|
UserColumnCode|RowLowRangeOrName|UserTableCode
MERGE|UserDefinedTable|VISION_UDT_MATCHED_ROW|M|T|Days Of The Week|VISION_UDT_MATCHED_ROW|US Legislative
 Data Group
```

```
MERGE|UserDefinedTableColumn|MONDAY_TO_THURSDAY|N|Monday_to_Thursday|VISION_UDT_MATCHED_ROW|US Legislative
 Data Group|
MERGE|UserDefinedTableColumn|TUESDAY_TO_FRIDAY|N|Tuesday_to_Friday|VISION_UDT_MATCHED_ROW|US Legislative
 Data Group|
MERGE|UserDefinedTableColumn|WEDNESDAY_TO_SATURDAY|N|Wednesday_to_Saturday|VISION_UDT_MATCHED_ROW|US
 Legislative Data Group|
MERGE|UserDefinedTableColumn|THURSDAY_TO_SUNDAY|N|Thursday_to_Sunday|VISION_UDT_MATCHED_ROW|US Legislative
 Data Group|
MERGE|UserDefinedTableRow|10|4712/12/31|2019/06/14||MONDAY|VISION_UDT_MATCHED_ROW|US Legislative Data Group|
Monday
MERGE|UserDefinedTableRow|20|4712/12/31|2019/06/14||TUESDAY|VISION_UDT_MATCHED_ROW|US Legislative Data
 Group|Tuesday
MERGE|UserDefinedTableRow|30|4712/12/31|2019/06/14||WEDNESDAY|VISION_UDT_MATCHED_ROW|US Legislative Data
 Group|Wednesday
MERGE|UserDefinedTableRow|40|4712/12/31|2019/06/14||THURSDAY|VISION_UDT_MATCHED_ROW|US Legislative Data
 Group|Thursday
MERGE|UserDefinedTableRow|50|4712/12/31|2019/06/14||FRIDAY|VISION_UDT_MATCHED_ROW|US Legislative Data Group|
Friday
MERGE|UserDefinedTableRow|60|4712/12/31|2019/06/14||SATURDAY|VISION_UDT_MATCHED_ROW|US Legislative Data
 Group|Saturday
MERGE|UserDefinedTableRow|70|4712/12/31|2019/06/14||SUNDAY|VISION_UDT_MATCHED_ROW|US Legislative Data Group|
Sunday
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|MONDAY_TO_THURSDAY|
MONDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|MONDAY_TO_THURSDAY|
TUESDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|MONDAY_TO_THURSDAY|
WEDNESDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|MONDAY_TO_THURSDAY|
THURSDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|MONDAY_TO_THURSDAY|
FRIDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|MONDAY_TO_THURSDAY|
SATURDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|MONDAY_TO_THURSDAY|
SUNDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|TUESDAY_TO_FRIDAY|
MONDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|TUESDAY_TO_FRIDAY|
TUESDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|TUESDAY_TO_FRIDAY|
WEDNESDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|TUESDAY_TO_FRIDAY|
THURSDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|TUESDAY_TO_FRIDAY|
FRIDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|TUESDAY_TO_FRIDAY|
SATURDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|TUESDAY_TO_FRIDAY|
SUNDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|
WEDNESDAY_TO_SATURDAY|MONDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|
WEDNESDAY_TO_SATURDAY|TUESDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|
WEDNESDAY_TO_SATURDAY|WEDNESDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|
WEDNESDAY_TO_SATURDAY|THURSDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|
WEDNESDAY_TO_SATURDAY|FRIDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|
WEDNESDAY_TO_SATURDAY|SATURDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|
WEDNESDAY_TO_SATURDAY|SUNDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|THURSDAY_TO_SUNDAY|
MONDAY|VISION_UDT_MATCHED_ROW
```

```
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|THURSDAY_TO_SUNDAY|
TUESDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|0|US Legislative Data Group|THURSDAY_TO_SUNDAY|
WEDNESDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|THURSDAY_TO_SUNDAY|
THURSDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|THURSDAY_TO_SUNDAY|
FRIDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|THURSDAY_TO_SUNDAY|
SATURDAY|VISION_UDT_MATCHED_ROW
MERGE|UserDefinedTableColumnInstance|4712/12/31|2019/06/14|10|US Legislative Data Group|THURSDAY_TO_SUNDAY|
SUNDAY|VISION_UDT_MATCHED_ROW
```

Related Topics

- Example of Loading User-Defined Tables for a Range of Row Values
- User Table Validation Formula Type

Examples of Loading Payroll Balance Definitions

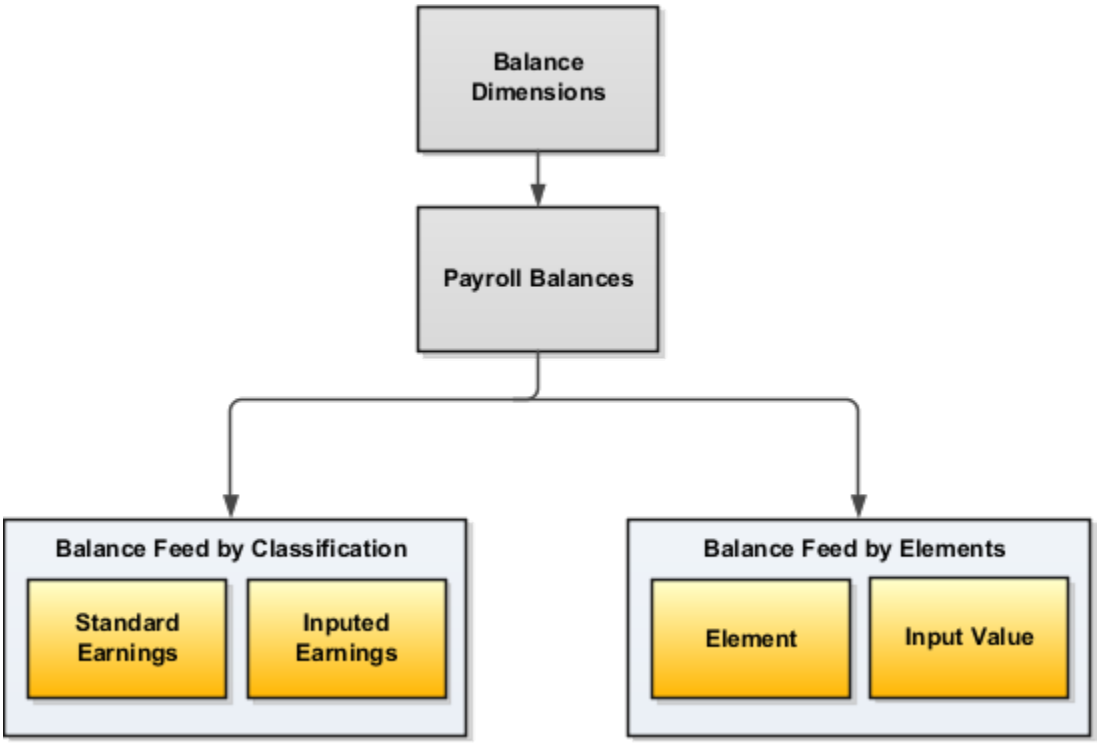
A balance definition includes a combination of user-defined naming criteria, one or more balance dimensions, and balance feeds. Most of the balances you require are predefined. Depending on your country extension, use HCM Data Loader to define additional balances for calculations or reporting.

When updating the seeded and custom balance feeds for payroll balance definitions, use separate files.

Let's consider this example. The members of Vision Corps union contribute their union dues that's determined by a formula. The formula computes the union dues as a percentage of a user-defined **Union Dues Basis** balance, which sums up these earnings.

- All earnings in the **Standard Earnings** classification.
- All earnings in the **Imputed Earnings** classification.
- A specific car allowance element.

This figure includes the balance dimensions that identify the specific value of a balance. Here, the balance is fed by the element classifications and an element.



Payroll balance definitions are made up of these components:

- Balances
- Defined Balance
- Balance Feed

Balances

Payroll balances show the accumulation of values over a period of time. Typically, the global and country-specific rules create a majority of balances whenever you create an earnings or deductions element. However, you might need to define a balance definition outside of the element setup.

You supply balance definitions in the `PayrollBalanceDefinition.dat` file. Here, you assign a category of **Miscellaneous** so that the balance isn't included in the reports. Since the monetary value is fed into the balance, you supply UOM as **Money** and the Currency as **USD**.

```
METADATA|PayrollBalanceDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|BalanceCode|
BalanceName|ReportingName|BalanceCategoryCode|BalanceUOM|CurrencyCode
MERGE|PayrollBalanceDefinition|VISION|UNION_DUES|Vision Corporation US LDG|HDL_UNION_DUES_BASIS|Union Dues
 Basis|Union Dues Basis|Miscellaneous|M|USD
```

Defined Balances

Each payroll balance can have multiple dimensions, which define the specific value to retrieve. The following Defined Balance lines load these relationship level dimensions for which the balance value is aggregated.

- Relationship Period to Date
- Relationship Month to Date
- Relationship Quarter to Date

- Relationship Year to Date

```
METADATA|DefinedBalance|SourceSystemOwner|SourceSystemId|BalanceId(SourceSystemId)|LegislativeDataGroupName|
BalanceDimensionCode
MERGE|DefinedBalance|VISION|UNION_DUES_DB_PTD|UNION_DUES|Vision Corporation US LDG|Relationship Period to
 Date
MERGE|DefinedBalance|VISION|UNION_DUES_DB_MTD|UNION_DUES|Vision Corporation US LDG|Relationship Month to
 Date
MERGE|DefinedBalance|VISION|UNION_DUES_DB_QTD|UNION_DUES|Vision Corporation US LDG|Relationship Quarter Year
 to Date
MERGE|DefinedBalance|VISION|UNION_DUES_DB_YTD|UNION_DUES|Vision Corporation US LDG|Relationship Year to Date
MERGE|DefinedBalance|VISION|UNION_DUES_DB_RUN|UNION_DUES|Vision Corporation US LDG|Relationship Run
```

Balance Feed

A balance feed contains details of how a given element input value contributes to a specific balance. You can load element classifications and individual elements to feed a balance.

You can't change either the first effective start date or last effective end date for an existing balance feed. Create balance feed objects with effective start dates on or before the start dates of other objects that refer to your balance feed.

Balance Feed by Classification

These balance classification lines create balance classifications, which determine the element classifications you add to or subtract from the balance. The direct run result value of every element in the classification feeds the balance. Here, you load **Standard Earnings** and **Imputed Earnings** element classifications with Add feed.

```
METADATA|BalanceClassification|SourceSystemOwner|SourceSystemId|BalanceId(SourceSystemId)|
LegislativeDataGroupName|ElementClassificationCode|AddSubtract
MERGE|BalanceClassification|VISION|UNION_DUES_BC_STD|UNION_DUES|Vision Corporation US LDG|Standard Earnings|
1
MERGE|BalanceClassification|VISION|UNION_DUES_BC_TAX|UNION_DUES|Vision Corporation US LDG|Taxable Benefits|
1
```

Balance Feed by Elements

Load individual elements and input values to feed the balance.

These balance feed lines add an element feed from the **ZFRC VS USD Car Allowance Element Earnings Results** element and **Earnings Calculated** input value.

```
METADATA|BalanceFeed|SourceSystemOwner|SourceSystemId|BalanceId(SourceSystemId)|EffectiveStartDate|
LegislativeDataGroupName|ElementCode|InputValueCode|AddSubtract
MERGE|BalanceFeed|VISION|UNION_DUES_BF_IMP|UNION_DUES|2018/01/01|Vision Corporation US LDG|ZFRC VS USD Car
 Allowance Element Earnings Results|Earnings Calculated|1
```

Related Topics

- Payroll Balance Definitions
- Balance Feeds

Payroll Balance Groups

Example of Loading Payroll Balance Attribute Definitions

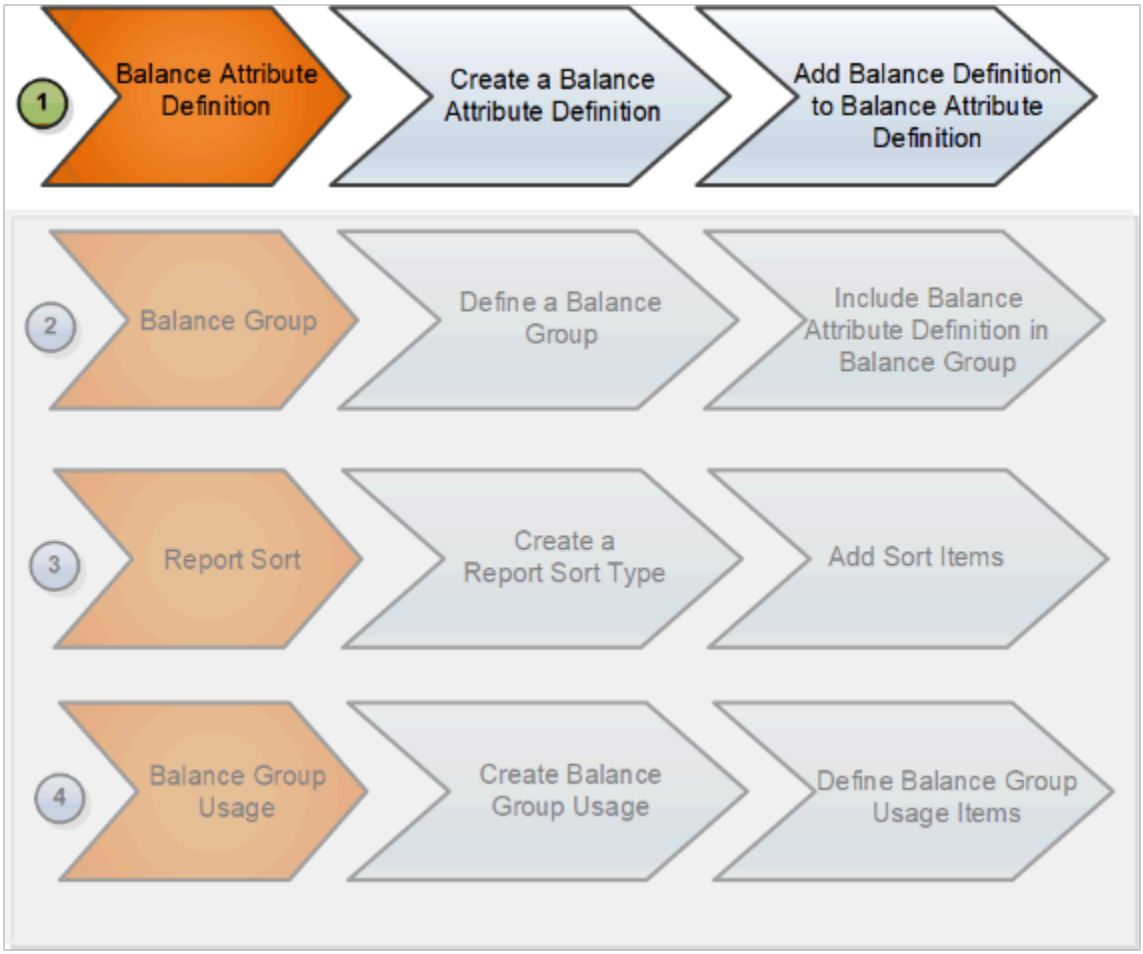
Use HCM Data Loader to define balance attribute definitions and add balance definitions to it.

A balance attribute definition is a set of balance definitions that you can include in your balance group. It identifies the criteria for including the balance definitions. Use a balance attribute to associate a balance definition to a balance attribute definition.

In this example, the Vision Corp Company wants to report a particular statutory deduction balance for employees. To do this, first you create a balance attribute definition Employee Taxes Attribute Definition and add these balance definitions to it.

Balance Type	Balance Dimension
FIT Withheld	Relationship Tax Unit Period to Date
Medicare Employee Withheld	Relationship Tax Unit Period to Date
Social Security Employee Withheld	Relationship Tax Unit Period to Date
FIT Withheld	Relationship Tax Unit Quarter to Date
Medicare Employee Withheld	Relationship Tax Unit Quarter to Date
Social Security Employee Withheld	Relationship Tax Unit Quarter to Date
FIT Withheld	Relationship Tax Unit Year to Date
Medicare Employee Withheld	Relationship Tax Unit Year to Date
Social Security Employee Withheld	Relationship Tax Unit Year to Date

As this figure shows, you create a balance attribute definition and then add balance definitions to it.



The next step is to create a balance group and add this balance attribute definition to the balance group. Later you create a report sort type to sort the balance values and then create a balance group usage for your balance group.

Creating a Balance Attribute Definition

In the **BalanceAttributeDefinition.dat** file you define a balance attribute definition and add balance definitions to it.

```
METADATA|BalanceAttributeDefinition|LegislativeDataGroupName|AttributeCode|UserAttributeName|Alterable|
Description
MERGE|BalanceAttributeDefinition|Vision Corporation US LDG|EMP_TAXES_ATTR_DEFN|Employee Taxes Attribute
 Definition|Y|Attribute definition for Employee Taxes reporting
METADATA|BalanceAttribute|LegislativeDataGroupName|AttributeCode|BalanceCode|BalanceDimensionCode
MERGE|BalanceAttribute|Vision Corporation US LDG|EMP_TAXES_ATTR_DEFN|US_FIT_WITHHELD|Core Relationship Tax
 Unit Period to Date
MERGE|BalanceAttribute|Vision Corporation US LDG|EMP_TAXES_ATTR_DEFN|US_FIT_WITHHELD|Core Relationship Tax
 Unit Quarter Year to Date
MERGE|BalanceAttribute|Vision Corporation US LDG|EMP_TAXES_ATTR_DEFN|US_FIT_WITHHELD|Core Relationship Tax
 Unit Year to Date
MERGE|BalanceAttribute|Vision Corporation US LDG|EMP_TAXES_ATTR_DEFN|US_MEDICARE_EMPLOYEE_WITHHELD|Core
 Relationship Tax Unit Period to Date
MERGE|BalanceAttribute|Vision Corporation US LDG|EMP_TAXES_ATTR_DEFN|US_MEDICARE_EMPLOYEE_WITHHELD|Core
 Relationship Tax Unit Quarter Year to Date
MERGE|BalanceAttribute|Vision Corporation US LDG|EMP_TAXES_ATTR_DEFN|US_MEDICARE_EMPLOYEE_WITHHELD|Core
 Relationship Tax Unit Year to Date
MERGE|BalanceAttribute|Vision Corporation US LDG|EMP_TAXES_ATTR_DEFN|US_SOCIAL_SECURITY_EMPLOYEE_WITHHELD|
Core Relationship Tax Unit Period to Date
```

```
MERGE|BalanceAttribute|Vision Corporation US LDG|EMP_TAXES_ATTR_DEFN|US_SOCIAL_SECURITY_EMPLOYEE_WITHHELD|
Core Relationship Tax Unit Quarter Year to Date
MERGE|BalanceAttribute|Vision Corporation US LDG|EMP_TAXES_ATTR_DEFN|US_SOCIAL_SECURITY_EMPLOYEE_WITHHELD|
Core Relationship Tax Unit Year to Date
```

Related Topics

- Example of Loading Payroll Balance Groups
- Example of Loading Report Sort Type and Report Sort Items
- Example of Loading Payroll Balance Group Usages
- Balance Groups
- Create Balance Groups and Usages

Example of Loading Payroll Balance Groups

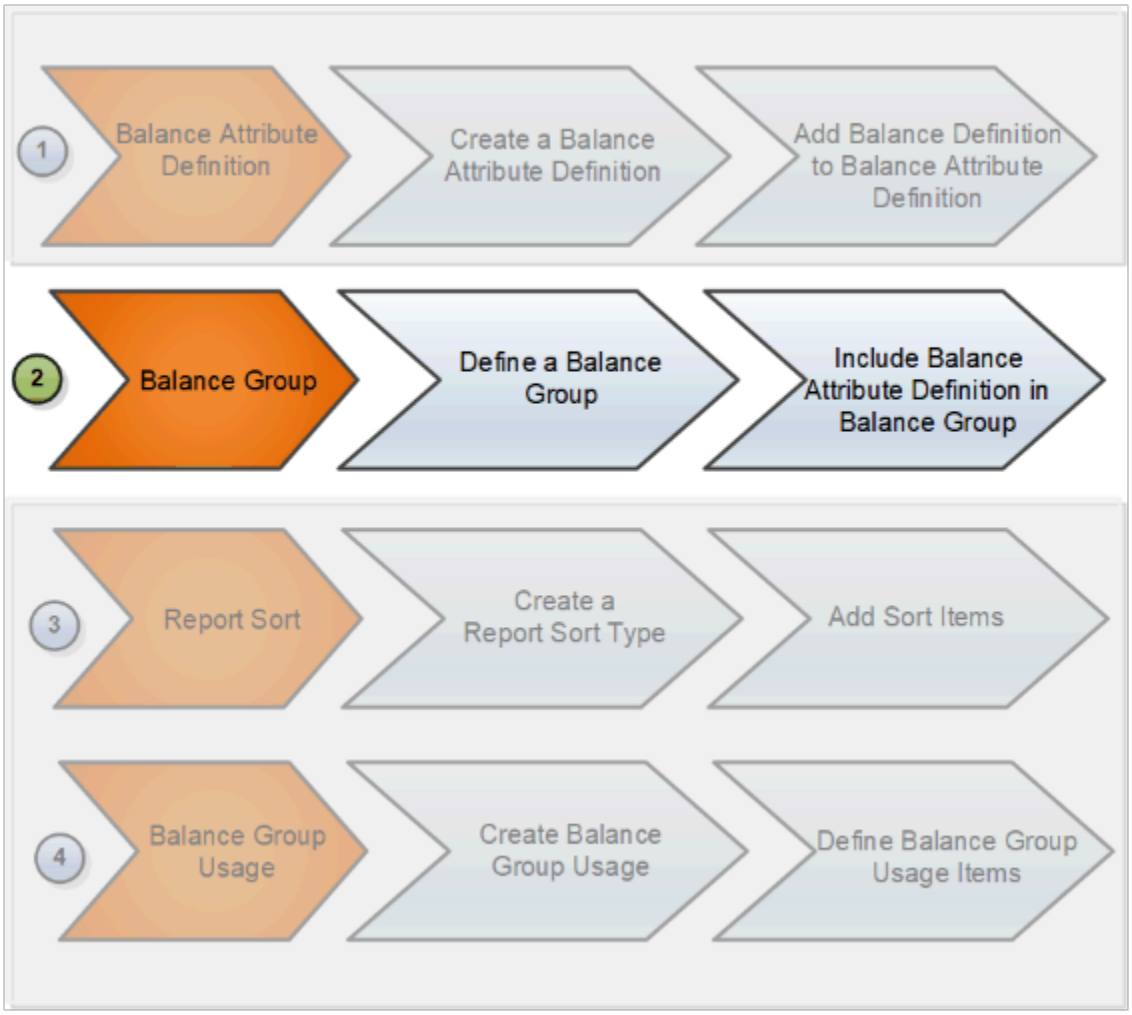
Use HCM Data Loader to define payroll balance groups for viewing and reporting payroll balances. You can define a balance group either at the employee level or at group level.

In this example, you define a payroll balance group **Employee Taxes Balance Group** and add the balance attribute definition to it.

Before you do this, ensure that you have created the balance attribute definition **Employee Taxes Attribute Definition**. Also, you must have added balances to the attribute definition.

Refer to the topic: Loading Payroll Balance Attribute Definitions for more information on creating a balance attribute definition.

As this figure shows, you define a balance group and later include the balance attribute definitions to it.



After defining a payroll balance group, the next step is to create a sort list to sort the balance items. Then you create a balance group usage and add this balance group to a balance group usage.

Creating a Payroll Balance Group

Use the **BalanceGroup.dat** file to create a payroll balance group and include the balance attribute definition in it.

```
METADATA|BalanceGroup|LegislativeDataGroupName|BalanceGroupCode|GroupName|Description|
BalanceCategoryRestricted|BalanceDimensionRestricted
MERGE|BalanceGroup|Vision Corporation US LDG|EMP_TAXES_BAL_GRP|Employee Taxes Balance Group|Balance group
 for Employee Taxes reporting|N|N
METADATA|BalanceGroupInclusion|LegislativeDataGroupName|BalanceGroupCode|AttributeCode
MERGE|BalanceGroupInclusion|Vision Corporation US LDG|EMP_TAXES_BAL_GRP|EMP_TAXES_ATTR_DEFN
```

Related Topics

- Example of Loading Payroll Balance Group Usages
- Example of Loading Report Sort Type and Report Sort Items
- Balance Groups
- Create Balance Groups and Usages
- Rules for Editing Balance Groups and Their Usages

Example of Loading Report Sort Type and Report Sort Items

Use HCM Data Loader to create a report sort type and items that defines how balance values are sorted in the report.

Let's look at this example. Vision Corp creates a report sort type and specifies the sort type as Static Order to define the sequence that displays the balance results.

Before You Start

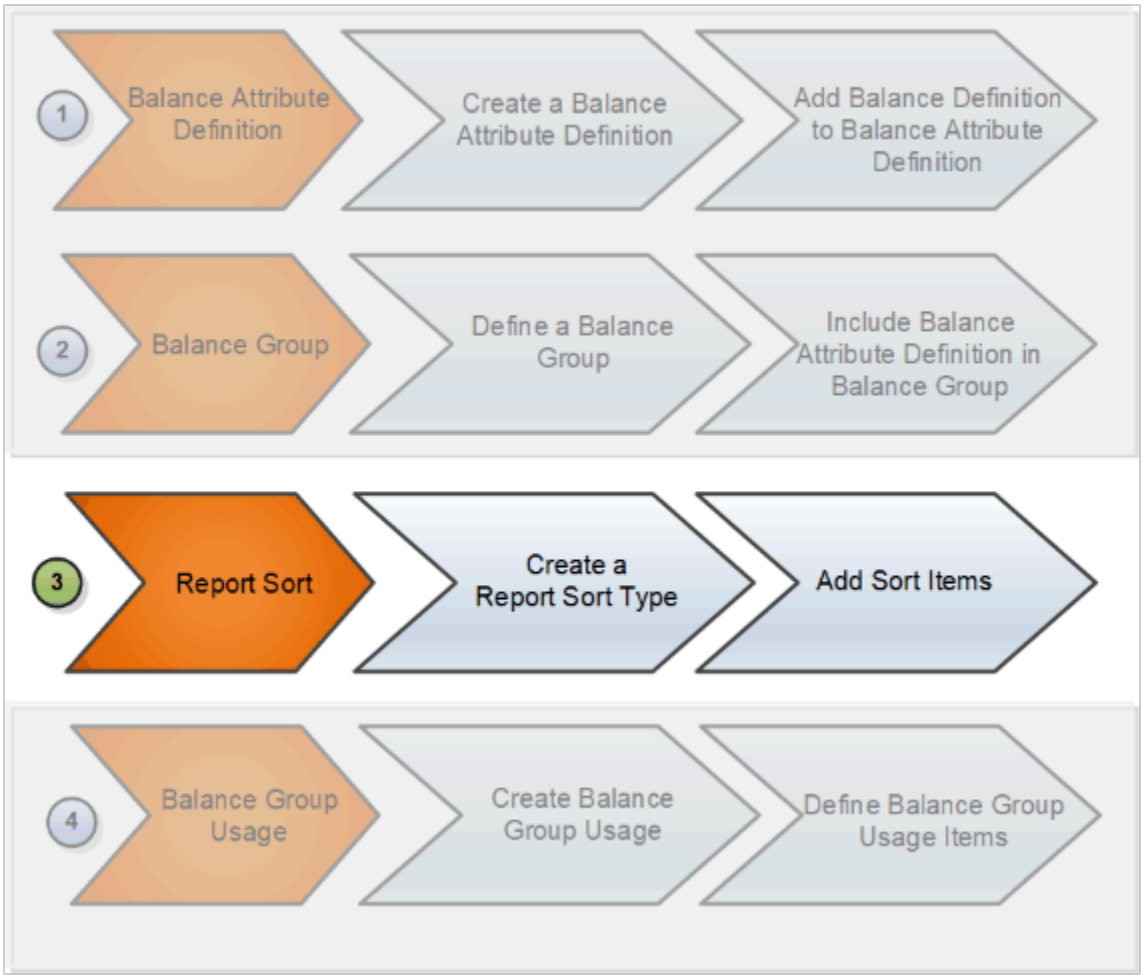
Ensure that you have completed these tasks:

- Created the balance attribute definition **Employee Taxes Attribute Definition**.

  For more information on how to create balance attribute definitions, refer to the topic: Loading Balance Attribute Definitions.
- Created the balance group **Employee Taxes Balance Group** and included the balance attribute definitions to it.

  For more information on how to create and load a balance group using HCM Data Loader, refer to the topic: Loading Payroll Balance Groups.

As this figure shows, you create a report sort type and add sort items to it.



After creating a report sort type, you create a balance group usage for your balance group and define the balance group usage items. For more information on how to create a balance group usage using HCM Data Loader, refer to the topic: Loading Payroll Balance Groups Usage.

Creating a Report Sort Type

Use the **ReportSort.dat** file to create a report sort type with sort method as Static Order.

```
METADATA|ReportSortType|LegislativeDataGroupName|ReportSortCode|SortName|Description|SortMethod|SortLevel|
SortOrder
MERGE|ReportSortType|Vision Corporation US LDG|EMP_TAXES_STATIC_SORT|Employee Taxes Sort|Static sort order
 for Employee Taxes|Static Order|Balance Type|Ascending
```

Creating Report Sort Items

Use the **ReportSortItem.dat** file to specify the order in which balance types are displayed for the balance group usage.

```
METADATA|ReportSortItem|LegislativeDataGroupName|ReportSortCode|SequenceNumber|BalanceCode
MERGE|ReportSortItem|Vision Corporation US LDG|EMP_TAXES_STATIC_SORT|10|US_FIT_WITHHELD
MERGE|ReportSortItem|Vision Corporation US LDG|EMP_TAXES_STATIC_SORT|20|US_MEDICARE_EMPLOYEE_WITHHELD
MERGE|ReportSortItem|Vision Corporation US LDG|EMP_TAXES_STATIC_SORT|30|US_SOCIAL_SECURITY_EMPLOYEE_WITHHELD
```

Related Topics

- Example of Loading Payroll Balance Group Usages
- Balance Groups
- Create Balance Groups and Usages
- Rules for Editing Balance Groups and Their Usages

Example of Loading Payroll Balance Group Usages

Use HCM Data Loader to create balance group usages that represent an instance of a balance group, such as report, extract definition, and so on.

For every balance group, you must define at least one balance group usage. But you can define multiple usages for a balance group.

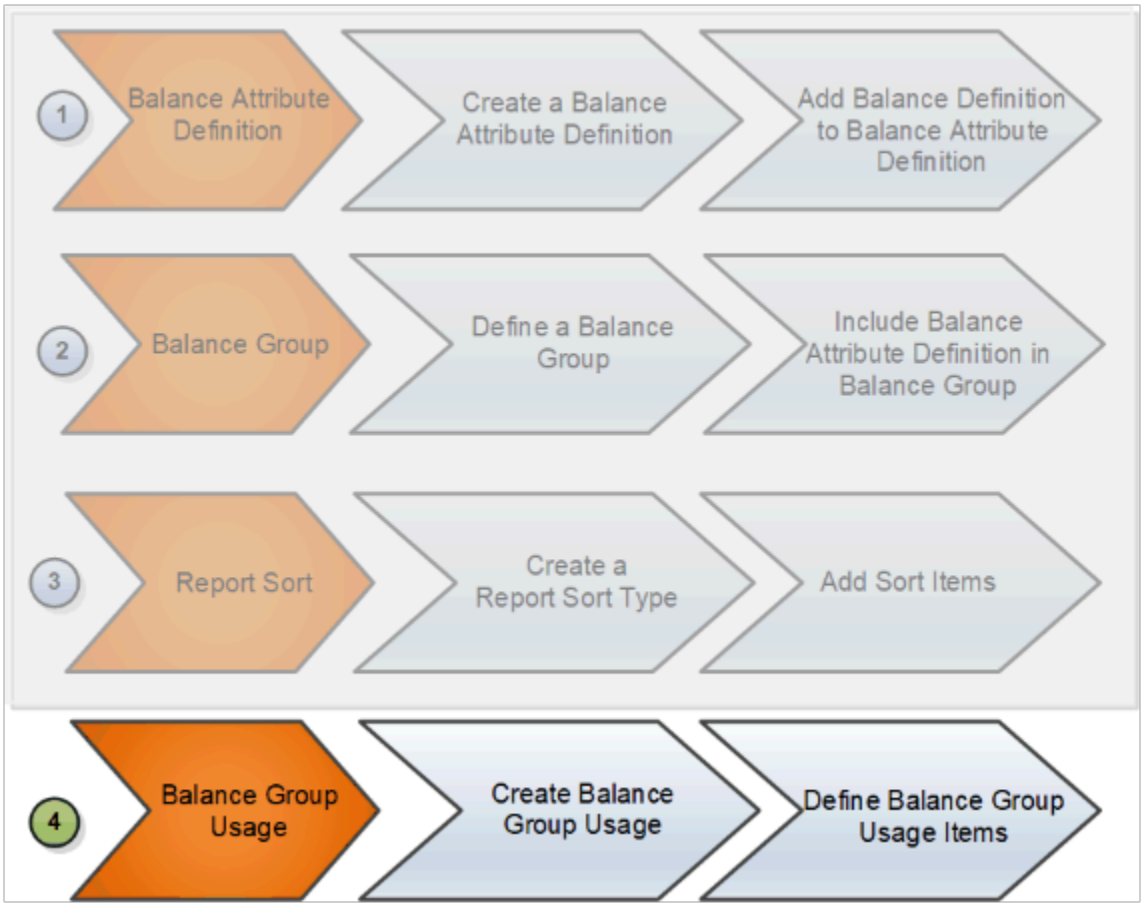
Vision Corp creates a balance group usage to use the **Employee Taxes Balance Group** balance group for their employee taxes report. Include your report sort type **Employee Taxes Sort** to this balance group usage.

Before You Start

Ensure that you have completed these tasks:

- Created the balance attribute definition **Employee Taxes Attribute Definition**. For more information on how to create balance attribute definitions, refer to the topic: Loading Balance Attribute Definitions.
- Created the balance group **Employee Taxes Balance Group** and included the balance attribute definitions to it. For more information on how to create and load a balance group using HCM Data Loader, refer to the topic: Loading Payroll Balance Groups.
- Created the Report Sort Type **Employee Taxes Sort** and added the sort items to it. For more information on how to create report sort type, refer to the topic: Loading Report Sort.

As this figure shows, you create a balance usage for the balance group and define the balance group usage items.



Creating a Payroll Balance Group Usage

This example **BalanceGroupUsage.dat** file creates the Employee Period Taxes Balance group Usage in a matrix format. Each balance group usage item defines a column in the Employee Period Taxes balance results matrix.

```
METADATA|BalanceGroupUsage|LegislativeDataGroupName|BalanceGroupUsageCode|GroupUsageName|Description|
FormatType|BalanceGroupCode|ReportSortCode
MERGE|BalanceGroupUsage|Vision Corporation US LDG|EMP_PERIOD_TAXES|Employee Period Taxes|Employee taxes
 matrix with static sort order|MATRIX|EMP_TAXES_BAL_GRP|EMP_TAXES_STATIC_SORT
METADATA|BalanceGroupUsageItem|LegislativeDataGroupName|BalanceGroupUsageCode|SourceType|Position|
BalanceDimensionCode
MERGE|BalanceGroupUsageItem|Vision Corporation US LDG|EMP_PERIOD_TAXES|Balance Dimension|10|Core
 Relationship Tax Unit Period to Date
MERGE|BalanceGroupUsageItem|Vision Corporation US LDG|EMP_PERIOD_TAXES|Balance Dimension|20|Core
 Relationship Tax Unit Quarter Year to Date
MERGE|BalanceGroupUsageItem|Vision Corporation US LDG|EMP_PERIOD_TAXES|Balance Dimension|30|Core
 Relationship Tax Unit Year to Date
```

Related Topics

- Balance Groups
- Create Balance Groups and Usages
- Rules for Editing Balance Groups and Their Usages

Fast Formulas

Example of Loading Fast Formula

Use HCM Data Loader to create formula that are generic expressions of calculations or comparisons that you want to repeat with different input variables. You can use these predefined rules to perform calculations, such as a legislative specific tax or social insurance calculations.

Create formula with effective start dates on or before the start dates of other objects that refer to your formula.

Supply the fast formula text in a separate text file and then pass the text file name to the Comments attribute. Further, place the formula text file in the `clobFiles` folder within the same compressed file as FastFormula.dat file that references it. Also, don't supply the text directly in the FastFormula.dat file.

This example loads the fast formula.

```
METADATA|FastFormula|SourceSystemOwner|SourceSystemId|EffectiveStartDate|FormulaCode|FormulaTypeCode|
FormulaName|FormulaText|LegislativeDataGroupId(SourceSystemId)
MERGE|FastFormula|VISION|MGR_SCHED_HRS|2000/01/01|MGR_RANGE_SCHD_HRS|Range of Scheduled Hours|Manager Range
 of Scheduled Hours|ManagerRangeScheduledHrs.txt|
```

Related Topics

- Overview of Using Formulas

Example of Loading Fast Formula Globals

Use fast formula globals in multiple formulas to support the Human Capital Management business rules.

For example, the global value could hold tax relief percentages, bonus percentage or overtime rates. Create globals with effective start dates on or before the formula that refer to them.

This example loads global values using source keys.

```
METADATA|FastFormulaGlobal|SourceSystemOwner|SourceSystemId|EffectiveStartDate|GlobalCode|GlobalName|
DataType|GlobalValue|LegislativeDataGroupId(SourceSystemId)
MERGE|FastFormulaGlobal|VISION|GBLMAXILLHEALTH|2000/01/01|MAX_ABS_DURATION_ILLHEALTH|Maximum Absence
 Duration for Ill Health|N|10|
MERGE|FastFormulaGlobal|VISION|GBLDATEEXAMPLE|2000/01/01|GLOBAL_DATE_EXAMPLE|Fast Formula Global Date
 Example|D|2012-01-01|LDG_VI_UK
MERGE|FastFormulaGlobal|VISION|GBLSTRINGEXAMPLE|2000/01/01|GLOBAL_STRING_EXAMPLE|Fast Formula Global String
 Example|T|Global Value|LDG_VI_UK
```

Create the legislative data group using the example provided in the Loading Legislative Data Groups: Examples topic.

Related Topics

- Overview of Using Formulas

Elements

Overview

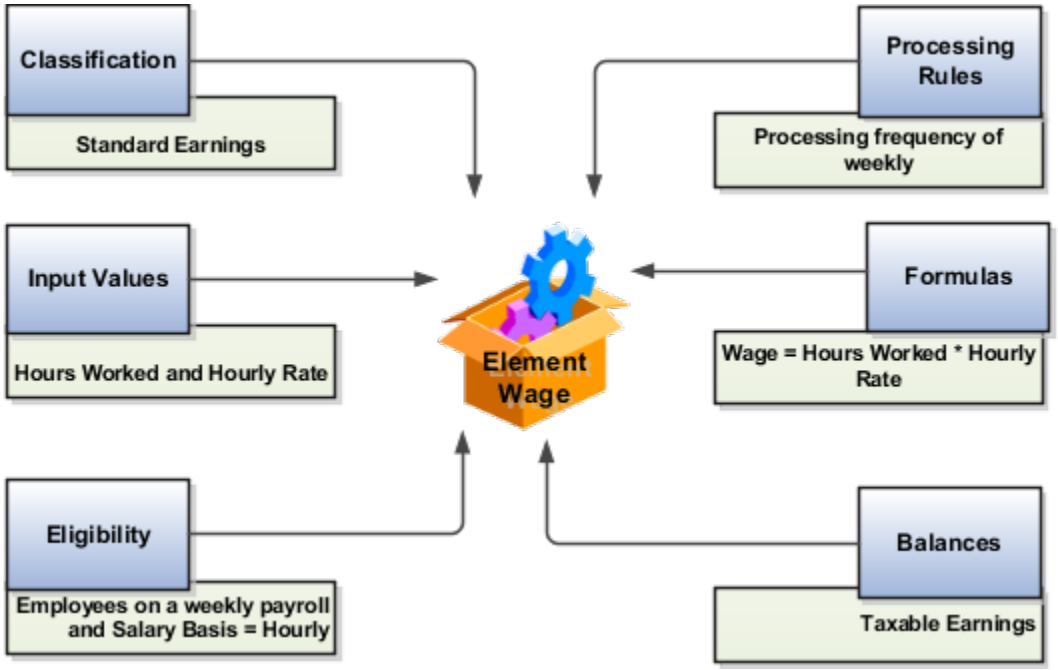
Use HCM Data Loader to load elements, which determine the payment of base pay, benefits, absences, earnings and deductions. For example, you can load earnings and deduction elements, such as bonuses, overtime earnings, and involuntary deductions.

You create elements by using the element templates. The design of the template varies depending upon the primary and secondary classifications and the element category. The answers that you load to the template questions determine the components that are created for the element, such as the balances, balance feeds, and formulas required for processing.

Refer to the Element Creation: Review page for a specific list of questions that you will need to answer to load your elements.

> **Note:** The application automatically submits a scheduled process to create the elements. The HCM Data Loader loads the data into template tables and in turn submits the scheduled process to create the element. The scheduled process performs any validations specific to element creation, and you must verify the log file for any errors. The HCM Data Loader may show the status as successful, but the actual element creation may fail because of the element template validations.

This figure illustrates an example of an earnings element for hourly wages. For each component, you will load the respective value. For example, you load Standard Earnings as the value for the component Classification.



This table summarizes the various building blocks of an element:

Component	What It Does
Classification	Determines the processing order and balance feeds for elements. Elements can belong to three types of classifications - Primary, Secondary, and Subclassifications.
Input Values	Defines values that hold information to calculate an element's payroll run result.
Eligibility	Determines rules that indicate which employees are eligible for an element.
Processing Rules	Identifies the formula to create and then use when processing element entries. You can load multiple processing rules if you want to use different formulas, such as a rule depending on an assignment status.
Formulas	Specifies the calculation used to process elements. An element has one or more formula result rules automatically generated when you create an element.
Balances	Accumulates result totals over a period of time.

Related Topics

- Payroll Elements
- Primary Element Classifications
- How Element Classification Components Work Together

Example of Creating Earnings Element for Payroll

This example shows how to use the element template to create a regular earnings element, such as salary. After you create the earnings element, you must create at least one eligibility record for it.

Load these details:

Parameter	Value
Legislative Data Group	Vision Corp
Primary Classification	Standard Earnings
Category	Standard
Element Name	Base Salary
Reporting Name	Base Salary
Effective Date	01/01/2010

Parameter	Value

The table shows the responses that you provide to these questions:

Question	Answer
What's the input currency?	US Dollar
Should every person eligible for the element automatically receive it?	No
What's the earliest entry date for this element?	First Standard Earnings Date
What's the latest entry date for this element?	Last Standard Earning Date
At which employment level should this element be attached?	Assignment Level
Do you want the element to be processed at Payroll Assignment level?|	Yes
Does the element recur each payroll period, or does it require explicit entry?	Recurring
Process the element only once in each payroll period?	Yes
Can a person have more than one entry of the element in a payroll period?	No
Process and pay element separately or with other earnings elements?	Process and pay with other earnings
Tax this earning across multiple pay periods?	No
Prorate this earning across all periods during which it was earned, and consider it for Overtime calculations, such as for commissions, bonuses, incentives, and other nondiscretionary earnings?	No
Does this element have a limit on the amount which is exempt from Federal Tax?|	No

Question	Answer
Does this element have a limit on the amount which is exempt at state level?	No
Do you want to default it to Federal level?	Yes
What's the calculation rule?	Flat Amount
What's the default periodicity of the element?	Periodically
What's the periodicity conversion rule?	Standard Rate Annualized
How do you want the work units to be reported?	Hours
What's the work units conversion rule?	Standard Rate Annualized
Is this element subject to proration?	Yes
What's the proration group?	Entry Changes for Proration
What's the proration rate conversion rule?	Periodic Work Schedule Rate Annualized
What's the proration units?	Hourly
Is this element subject to retroactive changes?	Yes
What's the retro group?	Entry Change for Retro
Use this element to calculate a gross amount from a specified net amount?	No
Should this element reduce regular earnings?	No
Should this element be included in the earnings calculation of the FLSA overtime base rate?	Yes
Should this element be included in the hours calculation of the FLSA overtime base rate?	Yes

Use this DAT file to create a regular earnings element.

```
METADATA|PayrollElementDetails|ElementName|ReportingName|Description|LegislativeDataGroupName|
PrimaryClassificationName|SecondaryClassificationName|Category|ElementStartDate
METADATA|PayrollElementQuestionnaire|ElementName|LegislativeDataGroupName|RuleCode|Rule|Response
MERGE|PayrollElementDetails|Base Salary|Base Salary|Base Salary|Vision Corporation US LDG|Standard
 Earnings||Standard|2010/01/01
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|SpecCurrency|Input Currency|US
 Dollar
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Availability Rule|Should every
 person eligible for the element automatically receive it?|No
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Hire Process|What is the earliest
 entry date for this element?|First Standard Earning Date
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Terminate Process|What is the latest
 entry date for this element?|Last Standard Earning Date
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Processing Level2Tier|At which
 employment level should this element be attached?|Assignment Level
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Processing Asg Level|Do you want the
 element to be processed at Payroll Assignment level?|Yes
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|RecurringOrNonRecurringExcl|Does
 this element recur each payroll period, or does it require explicit entry?|Recurring
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Once Per Period-Earnings|Process the
 element only once in each payroll period?|Yes
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Multiple Entries|Can a person have
 more than one entry of this element in a payroll period?|No
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Process Period|Process and pay
 element separately or with other earnings elements?|Process and pay with other earnings
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|AnnualizationRule|Tax this earning
 across multiple pay periods?|No
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|AugmentRule|Prorate this earning
 across all periods during which it was earned, and consider it for Overtime calculations, such as for
 commissions, bonuses, incentives, and other nondiscretionary earnings?|No
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|FedLimitRule|Does this element have
 a limit on the amount which is exempt from Federal Tax?|No
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|FedLimitRuleValue|Enter Federal Tax
 Limit|0
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|StateLimitRule|Does this element
 have a limit on the amount which is exempt at state level?|No
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|StateLimitRule2|Do you want to
 default it to Federal level?|Yes
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|CalculationRule|What is the
 calculation rule?|Flat amount
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|AugmentCalculationRule|What is the
 calculation rule?|Flat amount
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Periodicity|What is the default
 periodicity of this element?|Periodically
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|PeriodConverFormulas|Periodicity
 Conversion Rule|Standard Rate Annualized
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|ReportUnitRule|How do you want the
 work units to be reported?|Hours
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|RateConverFormula|Work Units
 Conversion Rule|Standard Rate Annualized
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Proration|Is this element subject to
 proration?|Yes
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Proration Group|Proration Group|
Entry Changes for Proration
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|ProrateRate|Proration Rate
 Conversion Rule|Periodic Work Schedule Rate Annualized
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|ProrationUnit|Proration Units|Hourly
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Retro|Is this element subject to
 retroactive changes?|Yes
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|RetroGroup|Retro Group|Entry Changes
 for Retro
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|IterativeInformation|Use this
 element to calculate a gross amount from a specified net amount?|No
```

```
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|Reduce Regular|Should this element
 reduce regular earnings?|No
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|FLSAEarnings|Should this element be
 included in the earnings calculation of the overtime base rate?|Yes
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|FLSAHours|Should this element be
 included in the hours calculation of the overtime base rate?|Yes
MERGE|PayrollElementQuestionnaire|Base Salary|Vision Corporation US LDG|AugmentFLSAEarnings|Should this
 element be included in the earnings calculation of the overtime base rate?|Yes
```

Related Topics

- Payroll Elements
- Primary Element Classifications
- How Element Classification Components Work Together

Example of Creating a Loan Deduction Element

In this example, an employee takes a personal loan and the Vision Corp company deducts a stipulated amount from their pay for each pay period.

When there are no sufficient earnings to cover the deduction, then the loan amount goes into the employee's arrears account.

Using HCM Data Loader, load these details:

Parameter	Value
Legislative Data Group	Vision Corp
Primary Classification	Voluntary Deduction
Category	Standard
Element Name	Personal Loan
Reporting Name	Personal Loan
Effective Date	01/01/2010

And this table shows the responses that you provide to these questions:

Question	Answer
What's the input currency?	US Dollar
Should every person eligible for the element automatically receive it?	No

Question	Answer
What's the earliest entry date for this element?	First Standard Earnings Date
What's the latest entry date for this element?	Last Standard Process Date
At which employment level should this element be attached?	Payroll relationship Level
Do you want the element to be processed at Payroll Relationship level?	Yes
What should happen when there are insufficient funds to cover the deductions?	Take a partial deduction, place remaining in arrears
Does the element recur each payroll period, or does it require explicit entry?	Recurring
Process the element only once in each payroll period?	Yes
Can a person have more than one entry of the element in a payroll period?	No
Do you want the element to be processed at Payroll Relationship level?	Yes
What's the calculation rule?	Flat amount deduction
Is this element subject to proration?	Yes
Proration Group	Entry Changes for Proration
Is this element subject to retroactive changes?	Yes
Retro Group	Entry Changes for Retro
Processing Stop when the Total is reached?	Yes

Use this DAT file to create a loan deduction element.

```
METADATA|PayrollElementDetails|ElementName|ReportingName|Description|LegislativeDataGroupName|
PrimaryClassificationName|SecondaryClassificationName|Category|ElementStartDate
```

```
METADATA|PayrollElementQuestionnaire|ElementName|LegislativeDataGroupName|RuleCode|Rule|Response
MERGE|PayrollElementDetails|Personal Loan|Personal Loan|Personal Loan|Vision Corporation US LDG|Voluntary
 Deductions||Standard|2010/01/01
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|SpecCurrency|Input Currency|US
 Dollar
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Availability Rule|Should every
 person eligible for the element automatically receive it?|No
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Hire Process|What is the earliest
 entry date for this element?|First Standard Earning Date
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Terminate Process|What is the
 latest entry date for this element?|Last Standard Process Date
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Processing Level|At which
 employment level should this element be attached?|Payroll relationship level
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Processing Rel Level|Do you want
 the element to be processed at Payroll Relationship level?|Yes
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Arrear Partial Dedn|What should
 happen when there are insufficient funds to cover the deductions?|Take a partial deduction, place remaining
 in arrears
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Recurring Or NonRecurring|Does
 this element recur each payroll period, or does it require explicit entry?|Recurring
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Once Per Period|Process the
 element only once in each payroll period?|Yes
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Multiple Entries|Can a person have
 more than one entry of this element in a payroll period?|No
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Processing Rel Level Excl|Do you
 want the element to be processed at Payroll Relationship level? Exclude|Yes
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|CalculationRuleDedn|What is the
 calculation rule?|Fixed amount deduction
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Proration|Is this element subject
 to proration?|Yes
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|ProrationFormula|Proration
 Formula|
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Proration Group|Proration Group|
Entry Changes for Proration
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Retro|Is this element subject to
 retroactive changes?|Yes
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|RetroGroup|Retro Group|Entry
 Changes for Retro
MERGE|PayrollElementQuestionnaire|Personal Loan|Vision Corporation US LDG|Total Reached|Processing Stop when
 the Total is reached?|Yes
```

Related Topics

- Payroll Elements
- Primary Element Classifications
- How Element Classification Components Work Together

Example of Deleting an Element

Use HCM Data Loader to delete elements. As you can't recover deleted records, you must take caution when deleting them.

The Payroll Manager of the Vision corp organization wants to create a regular earnings element for salary that recurs each payroll period. The Manager creates the element as nonrecurring instead of recurring. Recollect that after creating an element, they can't change the recurring entry attribute value from nonrecurring to recurring. So, they must delete the nonrecurring element that they wrongly created. Later, they can create the salary element afresh as recurring and with the same name.

This dat file deletes the element.

```
METADATA|PayrollElementDetails|ElementName|LegislativeDataGroupName
DELETE|PayrollElementDetails|VisionCorp_StandardEarnings|Vision Corporation US LDG
```

Related Topics

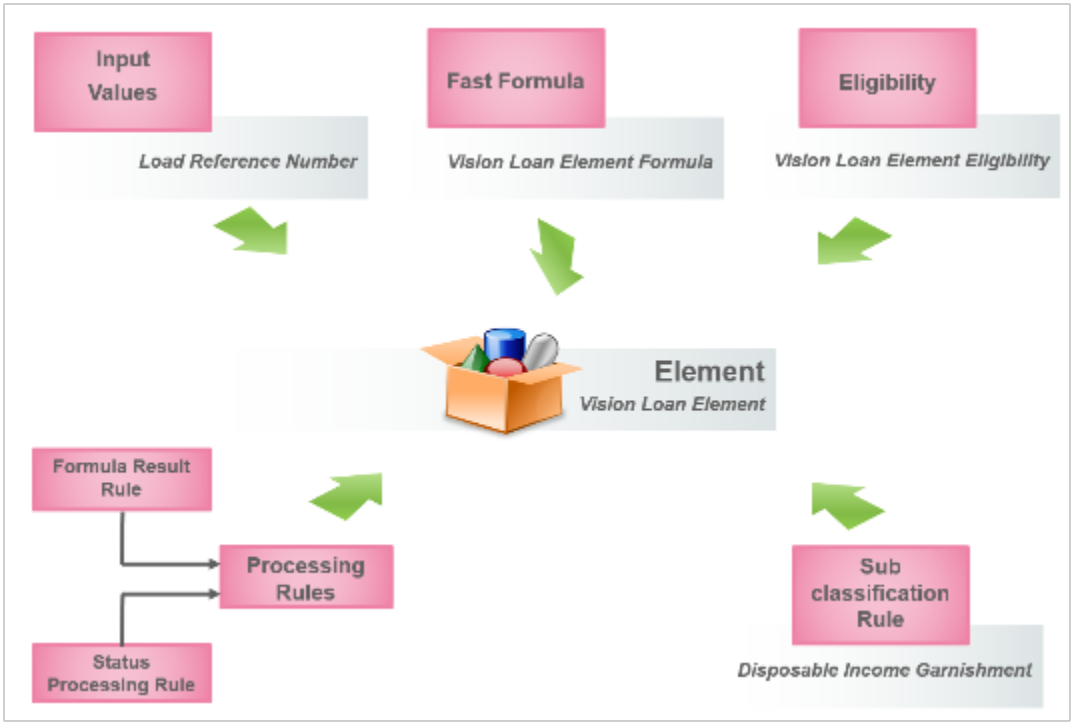
- Example of Creating Earnings Element for Payroll
- Example of Updating Payroll Element Definitions
- Example of Loading Payroll Element Run Usage
- Payroll Elements

Example of Updating Payroll Element Definitions

Use HCM Data Loader to create and update or delete existing element definitions. Element definitions include definitions for payroll, human resources, and benefits elements, such as earnings, deductions, and taxation.

You can edit the element definition to define the order in which the elements are processed in the payroll run. These rules determine whether the elements can be processed more than once in a period.

Let's consider the scenario where you define a new input value to an existing voluntary deduction element that doesn't have an element eligibility. You then create an eligibility, eligibility input values, and sub classification rules as shown in this figure.



These DAT lines update the existing element description and reporting name.

```
METADATA|PayrollElementDefinition|ElementCode|EffectiveStartDate|EffectiveEndDate|LegislativeDataGroupName|
ReportingName|Description
```

```
MERGE|PayrollElementDefinition|Vision Loan Element|2010/01/01|| Vision USA LDG|Car Loan Element|Process car
 loan deduction
```

Input Value

An element's input values define the entry values available on each entry of this element. Each input value has a unit of measure, such as money or date. Input values can include validations and conditions to control the data entry of the element entry assigned to a person.

These DAT lines create an input value **Loan Reference Number** for the element **Vision Loan Element**.

```
METADATA|InputValue|EffectiveStartDate|EffectiveEndDate|InputValueCode|ValueRequiredFlag|
CreateDatabaseItemFlag|UOM|DisplayFlag|AllowUserEntryFlag|SourceSystemOwner|SourceSystemId|ElementCode|
DisplaySequence|LegislativeDataGroupName|Name|ApplyDefaultAtRuntimeFlag
MERGE|InputValue|2010/01/01||Loan Reference Number|Y|Y|C|Y|Y|||Vision Loan Element|100| Vision USA LDG|Loan
 Reference Number|N
```

Element Eligibility

Element eligibility determines which people are eligible for an element. To determine eligibility, you select the criteria that people must have to receive entries of the element.

These DAT lines create the element eligibility Vision Loan Element Eligibility with the criteria Payroll = Vision Monthly Payroll.

```
METADATA|ElementEligibility|EffectiveStartDate|EffectiveEndDate|ElementEligibilityName|AutomaticEntryFlag|
ElementCode|LegislativeDataGroupName|PayrollCode
MERGE|ElementEligibility|2010/01/01||HDL Loan Element Eligibility|N|HDL Loan Element|PM US Sun Power|HDL
 Monthly Payroll
```

You can change the automatic entry attribute in the correction mode. All date-effective records are updated with this change.

You can end date the element eligibility to prevent the creation of further element entries. The example .dat file end dates the **'Vision_Monthly_Eligib'** element eligibility as on the effective date 31-Dec-2020.

```
METADATA|ElementEligibility|EffectiveStartDate|EffectiveEndDate|ReplaceLastEffectiveEndDate|
ElementEligibilityName|ElementCode|LegislativeDataGroupName
MERGE|ElementEligibility|2010/01/01|2020/12/31|Y|HDL Loan Element Eligibility|HDL Loan Element|PM US Sun
 Power
```

Formulas

Formulas specify the calculation used to process elements. Define a fast formula Vision Loan Element Formula that calculates the amount using various criteria. It returns the amount to a target element that's used to deduct the amount for an employee.

You associate the fast formula in the status processing rules and the deduction amount to the target element in formula result rules.

Status Processing Rule

Define rules for processing elements according to specific assignment statuses. For each assignment status, you can specify a different formula to be run for the same element.

These lines creates a status processing rule.

```
METADATA|StatusProcessingRule|EffectiveStartDate|EffectiveEndDate|ElementCode|LegislativeDataGroupName|
AssignmentStatusCode|FormulaCode|BalanceAdjustment
```

```
MERGE|StatusProcessingRule|2010/01/01||HDL Loan Element|PM US Sun Power|Inactive - No Payroll|HDL LOAN
 ELEMENT FORMULA|No
```

Formula Result Rule

A formula result rule defines the processing rules for each element. Formulas are attached to an element for processing according to specific assignment statuses. For each assignment status, it's possible to specify a different formula to be fired for the same element. For example, salary is calculated differently when the employee is on leave of absence.

These lines create a formula result rule.

```
METADATA|FormulaResultRule|EffectiveStartDate|EffectiveEndDate|ElementCode|LegislativeDataGroupName|
AssignmentStatusCode|ResultReturned|ResultRule|TargetElementCode|TargetInputValueCode|BalanceAdjustment
MERGE|FormulaResultRule|2010/01/01||HDL Loan Element|PM US Sun Power|Inactive - No Payroll|L_VALUE|Indirect
 Result|SC_HDL_EARN|Pay Value|No
```

Sub Classification Rule

Define sub classification rules for the elements to control the balance it feeds. An element can have multiple subclassifications.

These lines create a sub classification rule **Disposable Income Garnishment** for the **Standard Earnings** element.

```
METADATA|SubClassificationRule|EffectiveStartDate|EffectiveEndDate|ElementCode|LegislativeDataGroupName |
ClassificationCode
MERGE|SubClassificationRule|2010/01/01|| Disposable Income Garnishment | Vision USA LDG|Standard Earnings
```

Related Topics

- Example of Loading Fast Formula
- Payroll Elements
- How Element Classification Components Work Together
- Formula Result Rules for Elements
- Payroll Element Input Values

Example of Loading Payroll Element Run Usage

Use the payroll element run usage object to identify how an element is used with the run type. Run types control the elements and payment types to process in a payroll run.

You can't change either the first effective start date or last effective end date for an existing payroll element run usage object. Create these objects with effective start dates on or before the start dates of other objects that refer to your payroll element run usage.

Vision Corp processes and pays the bonus amounts separately from the regular earnings. The organization pays regular payroll by EFT and issues check bonuses once a year. For this requirement, they create a separate payment for the bonus element entry, which is marked to pay separately. Using HCM Data Loader, the Payroll Manager creates a bonus element, Annual Bonus, with Supplemental Earnings classification. She supplies these details.

Element Detail	Value
Primary Classification	Supplemental Earnings

Element Detail	Value
Secondary Classification	Bonus
Category	Standard
Process and pay element separately or with other earnings elements?	Process separately and pay separately

Further, she loads the run type usage as **Separate Payment** with Element Usage as **Trigger**. The application processes this element as well as makes payments separately.

This example dat file loads the payroll element run usage object:

```
METADATA|PayrollElementRunTypeUsage|EffectiveStartDate|UsageType|ElementTypeCode|RunTypeCode|
LegislativeDataGroupName
MERGE|PayrollElementRunTypeUsage|2018/02/02|Trigger| ANNUAL_BONUS|Regular|Vision Corporation US
```

Related Topics

- Payroll Elements
- Run Types
- How Pay Frequency Components Work Together

Loading Gross-up Balance Exclusion

Create a net-to-gross element using the element template. After you create the element, using HCM Data Loader, create the Gross Balance Exclusion to select the FIT Withheld balance.

When the Bonus element is processed in a payroll run, the application doesn't include the FIT Withheld in the gross-up calculation for Bonus.

Using the Create Element task, create a Bonus element as a supplemental earning and select these values as shown in the table:

Page	Question	Answer
Basic Information	Process and pay element separately or with other earnings elements	Process and pay with other earnings
Additional Details	Use this element to calculate a gross amount from a specified net amount?	Yes

Use this DAT file to create the gross balance exclusion using HCM Data Loader. You'll notice that on the Gross Balance Exclusions section of Element Summary, the Exclude Balances check box is selected for the FIT Withheld balance.

```
METADATA|PayrollNetToGrossBalanceExclusion|LegislativeDataGroupName|StartDate|EndDate|ElementCode|
BalanceCode|SourceSystemId|SourceSystemOwner
```

```
MERGE|PayrollNetToGrossBalanceExclusion|ZHRX_USVS_ST LDG One|2018/01/01|2020/12/31|Supp_Bonus_2 Earnings
 Results|US_FIT_WITHHELD|GBE0000001|HCMQA-001
```

Rate Definitions

Rate Definition Overview

Use HCM Data Loader to create rate definitions, such as compensation rates, element entry values, or rates based on values defined by criteria.

You can use rates to calculate payments that are based on different frequencies, such as hourly rates or annual rates. You can also define rates that are paid based on a number of units, such as meal allowance or mileage rates.

If the rate is based on multiple balance or element entries, or if it references other rate definitions, you can specify multiple rate contributors.

Categories

You can create a rate definition of one of these categories type:

Category	Used To
Derived Rate	Define rates based on values from one or more payroll balances or other rate definitions. Use this option to create a rate that retrieves a value from one or more rate contributors.
Element	Retrieve a value from or posts to an element input value. **Note:** Hours * Rates elements aren't supported. You can create a flat amount element to capture the hourly value. You can then use an element rates to calculate a rate stored on an hourly values for the employee, such as salary.
Formula	Define rates based on Fast Formula. Let's consider an example where an employee can work in a job at a higher grade than their normal job. The rate paid to that employee is based on different criteria and includes rules, such as the minimum rate must be at least 1.00. You can use a formula to step up the grade to the next level and to return the rate assigned to that higher grade. **Note:** Use this category for rate calculations in the payroll run. It isn't suitable for other types of rates such as salary rates or rates displayed on a time card.
Grade Rate	Calculate rates based on employee's grade details. For example, you can calculate the annual salary rate based on the grade ladder, grade, and grade step information defined for an employee.
Value by Criteria	Specify one or more evaluation conditions that determine a particular value or rate. For example, you define a value defined by criteria to capture car allowance, housing allowance and market supplement values based on an employee's job.

Category	Used To

Example of Loading Rate Definitions for Basic Salary

Use HCM Data Loader to create a primary rate for a basic salary. The salary rate for an employee is displayed on the salary page or used for other purposes, such as a contributor to an absence rate.

Before You Begin

Use HCM Data Loader to create the salary element at the assignment level. The salary element entries contain the information to be retrieved by the salary rate definition for an employee. You can create it using the flat amount or factor calculation rule.

Use this RateDefinitionV2.dat file to create a rate definition and load these values:

Attribute	Value
Category	Element
StorageType	Amount
ElementCode	Regular Salary
ReturnRatePeriodicity	Annually
ConversionFormulaCode	Standard Rate Annualized
ReturnRateCurrency	USD
LegislativeDataGroupName	CRFL RRF LDG US1

```
METADATA|RateDefinitionV2|RateDefinitionCode|ConversionFormulaCode|ContributorTotalPeriodicity|
ReturnRatePeriodicity|EffectiveStartDate|LegislativeDataGroupName|ShortName|Category|SourceSystemId|
SourceSystemOwner|ReturnRateCurrency|ElementCode|Status|BaseSalaryFlag|ReturnDecimalDisplay|
ReturnRoundingRule|StorageType|OverrideAllowed|ReturnFullTimeRateFlag|ReportingRateFlag|
CalculateLiveRatesFlag|Description|RateDefinitionName

MERGE|RateDefinitionV2|Basic_Salary|Standard Rate Annualized|Annually|Annually|2020/01/01|US LDG|
Basic_Salary|ET|RateElement_001|VISION|USD|Regular Salary|A|Y|6|D|A|Y|Y|Y|Y|Basic_Salary|Basic_Salary

METADATA|RateContributorV2|EmploymentLevel|ContributorType|ContributorInputValueCode|EffectiveStartDate|
Feed|LegislativeDataGroupName|RateDefinitionCode|SequenceNumber|ReturnFullTimeRateFlag|SourceSystemId|
SourceSystemOwner

MERGE|RateContributorV2|Payroll Assignment|IV|Amount|2020/01/01|1|US LDG|Basic_Salary|1|Y|RateElement_001|
VISION
```

Related Topics

- Rates Based on Retained Grades
- Create Rate Definitions for Leave
- Generate HCM Rates

Rates Based on Grades Details

You can calculate rates based on employee's grade details. For example, you can calculate the annual salary rate based on the grade ladder, grade, and grade step information defined for an employee.

When you define a rate definition, you can select a grade rate as a basis for the rate calculation and specify the employment level. The default employment level is Payroll Assignment. The application retrieves the grade ladder details based on assigned grade that's held on an employee's assignment record.

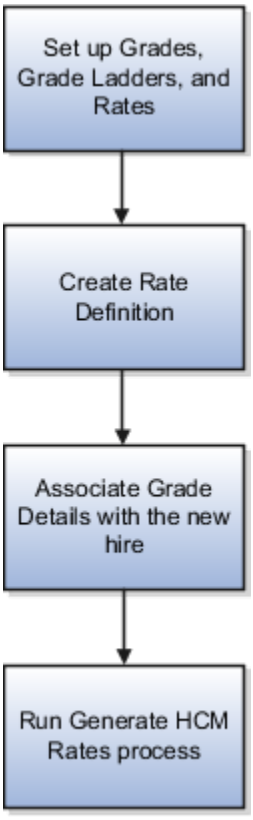
In your rate definition, you add either a rate contributor of type Grade Ladder or Retained Grade.

Calculate Rates Based on Grades

Follow these steps to calculate rates based on grades.

1. Set up grades, grade ladder, and rates for the given job to record the level of compensation for the employees.
2. Create a rate definition based on grade details. Add a rate contributor of type **Grade Ladder**.
3. Hire the employee and provide the required grade and step details.
4. Run the **Generate HCM Rates** process to calculate the rates based on the information held in the grade tables.

This flowchart describes the overall steps to calculate rates based on grade details.



Related Topics

- Example to Calculate Rates Based on Grades With Steps
- Example to Calculate Rates Based on Grades Without Steps
- Example to Calculate Rates Based on Grade Ladder With Multiple Assignments

Example to Load Rates Based on Grades With Steps

You hire Sherry Callaway as a Pharmacy Technician with grade 3 and step 2. Using HCM Data Loader, create a rate to calculate grade rate values for her annual salary payments based on grades, grade ladder, and step details.

Configuration

Follow these steps to calculate rates based on grades.

1. Set up grades, grade ladder, and rates for the **Pharmacy Technician** job to record the level of compensation for the technicians.
2. Create a rate definition based on grade details. Add a rate contributor of type **Grade Ladder**.
3. Hire Sherry in the grade 3 with Step 2.
4. Run the Generate HCM Rates Process to calculate the rates based on the information held in the grade tables.

Step 1: Grades, Grade Rates, and Grade Ladders

To set up the grade structure for the Pharmacy Technician job, perform these tasks:

1. Using HCM Data Loader, set up five different grades and add five steps for each grade.

   As shown in this illustration, set up grades 1 to 5. And add the five grade steps, Steps 1 through Step 5, to each of the grades.

   For more information on how to load grades, grade ladders, and grade rates, refer to these topics:

   - Guidelines for Loading Grades
   - Guidelines for Loading Grade Ladders
   - Guidelines for Loading Grade Rates

2. Load the Progression Grade Ladder object to create a grade ladder Pharmacy Technician Salary with these details:

Attribute	Value
Grade Set	Common Set
Name	Pharmacy Technicians Salary
Legislative Data Group	US Legislative Data Group
Frequency	Annually
Assignment Action	Automated Grade Step Progression

- Load the grades with steps in the sequence in which your employees typically progress in your organization. Enter the sequence for the place of the grade on the grade ladder, with 1 being the lowest grade. The employee can move up till the grade 5.

Grade	Step 1	Step 2	Step 3	Step 4	Step 5
1	17803	18398	18990	19579	20171
2	20017	20493	21155	21717	21961
3	21840	22568	23296	24024	24752
4	24518	25335	26152	26969	27786

Grade	Step 1	Step 2	Step 3	Step 4	Step 5
5	27431	28345	29259	30173	31087

For more information on how to load grade ladders, refer to these topics:

- Guidelines for Loading Progression Grade Ladders
- Examples of Loading Progression Grade Ladders with Steps

Step 2: Rate Definition

Load a rate definition of category type Grade Rate with these details:

Attribute	Value
ReturnRatePeriodicity	Annually
ConversionFormulaCode	Standard Rate Annualized
ReturnRateCurrency	USD

Using HCM Data Loader, use this RateDefinitionV2.dat file to load a rate contributor of type Grade Ladder.

```
METADATA|RateDefinitionV2|RateDefinitionCode|ConversionFormulaCode|ReturnRatePeriodicity|EffectiveStartDate|
LegislativeDataGroupName|ShortName|Category|SourceSystemId|SourceSystemOwner|ReturnRateCurrency|
Status|ReturnDecimalDisplay|ReturnRoundingRule|ReturnFullTimeRateFlag|ReportingRateFlag|Description|
RateDefinitionName

MERGE|RateDefinitionV2|Rate Def_Grade_Ladder|Standard Rate Annualized|Annually|2020/01/01|US LDG|Rate
 Def_Grade_Ladder|GRADE|RateGradeLadder_001|VISION|USD|A|6|D|Y|Y|Rate Def_Grade_Ladder|Rate Def_Grade_Ladder

METADATA|RateContributorV2|EmploymentLevel|ContributorType|EffectiveStartDate|Feed|LegislativeDataGroupName|
Periodicity|RateDefinitionCode|SequenceNumber|ReturnFullTimeRateFlag|SourceSystemId|SourceSystemOwner

MERGE|RateContributorV2|Payroll Assignment|GRLADDER|2020/01/01|1|US LDG|Annually|Rate Def_Grade_Ladder|1|Y|
RateGradeLadder_001|VISION
```

Step 3: Grade Details During Employee Hire

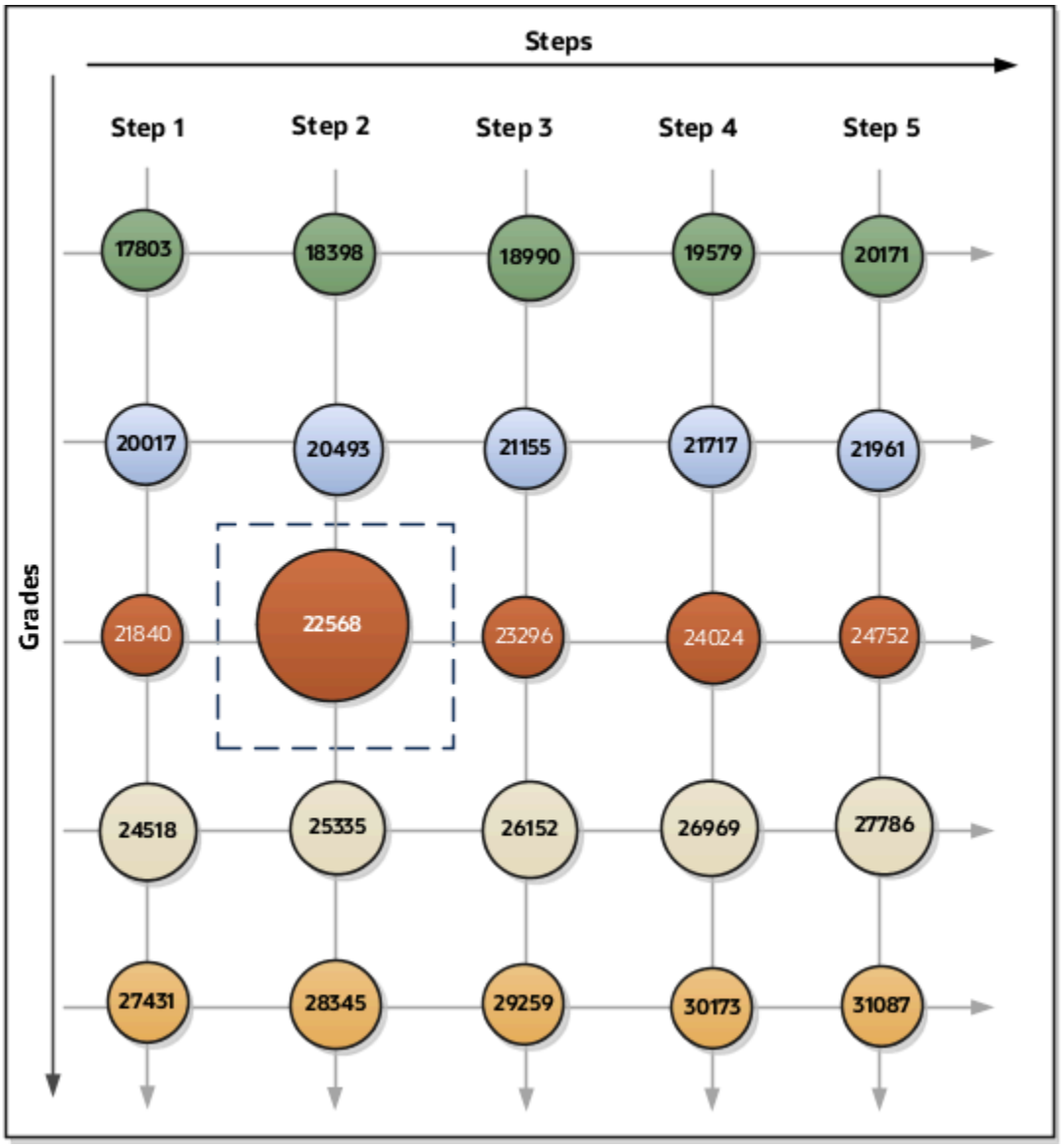
When you're hiring a new employee, add these grade ladder and grade details.

Attribute	Value
Grade Ladder	Pharmacy Technician Salary
Grade	3
Step	Step 2

Step 4: HCM Rates Process

On the Home page, click the **Payroll Flow Patterns** quick action under the **My Client Groups** tab. On the Payroll Flow Patterns page, after selecting a legislative data group, search for and submit the **Generate HCM Rates flow**.

Result: As shown in this illustration, the application returns a rate value of 22568.



Related Topics

- Guidelines for Loading Grades
- Guidelines for Loading Grade Ladders
- Guidelines for Loading Grade Rates
- Guidelines for Loading Progression Grade Ladders
- Examples of Loading Progression Grade Ladders with Steps

Load Rate Definitions for Overall Salary

Using HCM Data Loader, you load a rate definition for overall salary that includes multiple rate contributors.

You do these actions.

- Load the overall salary rate definition
- Add the regular salary rate contributor
- Add the car allowance rate contributor

This table summarizes the key decisions for your scenario.

Decision to Consider	In This Example
What components of pay should be included in an employee's overall salary?	• Regular Salary • Car Allowance
Should I include all pay for car allowance in the overall salary?	No. Only include 50 percent of the amount paid for car allowance.

Overall Salary

Using HCM Data Loader, you load the overall salary rate definition as shown in this table:

Attribute	Value
Category	Derived Rate
RateDefinitionName	Overall Salary
ElementCode	Salary
Periodicity	Weekly
ConversionFormulaCode	Standard Rate Annualized
ReturnedRateCurrency	USD
LegislativeDataGroupName	US LDG

Regular Salary Rate Contributor

This table provides the values that you provide when loading the regular salary rate contributor.

Attribute	Value
Add or Subtract	Add
Employment Level	Payroll Assignment
Periodicity	Weekly

Car Allowance Rate Contributor

This table provides the values that you provide when loading the car allowance rate contributor.

Attribute	Value
Add or Subtract	Add
Rate Name	Car Allowance
Periodicity	Weekly
Factor Rule	Value
Factor Value	0.5

This RateDefinitionV2.dat file loads the overall salary rate definition and the two contributors - Regular Salary and Car Allowances.

```
METADATA|RateDefinitionV2|RateDefinitionCode|ConversionFormulaCode|ContributorTotalPeriodicity|
ReturnRatePeriodicity|EffectiveStartDate|LegislativeDataGroupName|ShortName|Category|SourceSystemId|
SourceSystemOwner|ReturnRateCurrency|ElementCode|Status|ReturnDecimalDisplay|ReturnRoundingRule|
OverallSalaryFlag|ReturnFullTimeRateFlag|ProcessContributorTotalFteFlag|ReportingRateFlag|EmploymentLevel|
CalculateLiveRatesFlag|Description|RateDefinitionName

MERGE|RateDefinitionV2|Rate def_DRT_OverallSal|Standard Rate Annualized|Weekly|Weekly|2020/01/01|US LDG|
Rate def_DRT_OverallSal|DRT|RateElement_DRT_001|VISION|USD|AL_CDRM_Element_Rate|A|8|U|Y|Y|Y|Y|Payroll
 Assignment|N|Rate def_DRT_OverallSal|Rate def_DRT_OverallSal
METADATA|RateContributorV2|ContributorType|ContributorRateDefinitionCode|EffectiveStartDate|
Feed|LegislativeDataGroupName|Periodicity|RateDefinitionCode|SequenceNumber|RateContributorId|
ReturnFullTimeRateFlag|SourceSystemId|SourceSystemOwner

MERGE|RateContributorV2|RD|Basic_Salary|2020/01/01|1|US LDG|Weekly|Rate def_DRT_OverallSal|1||Y|
RateCont_DRT_030|VISION

MERGE|RateContributorV2|RD|Rate Def_Factor_Value|2020/01/01|1|US LDG|Weekly|Rate def_DRT_OverallSal|2||Y|
RateCont_DRT_031|VISION
```

Related Topics

- Rates Based on Grades Details
- Example to Load Rates Based on Grades With Steps
- Load Rate Definitions for Overall Salary

Values Defined by Criteria

Overview of Loading Values Defined By Criteria

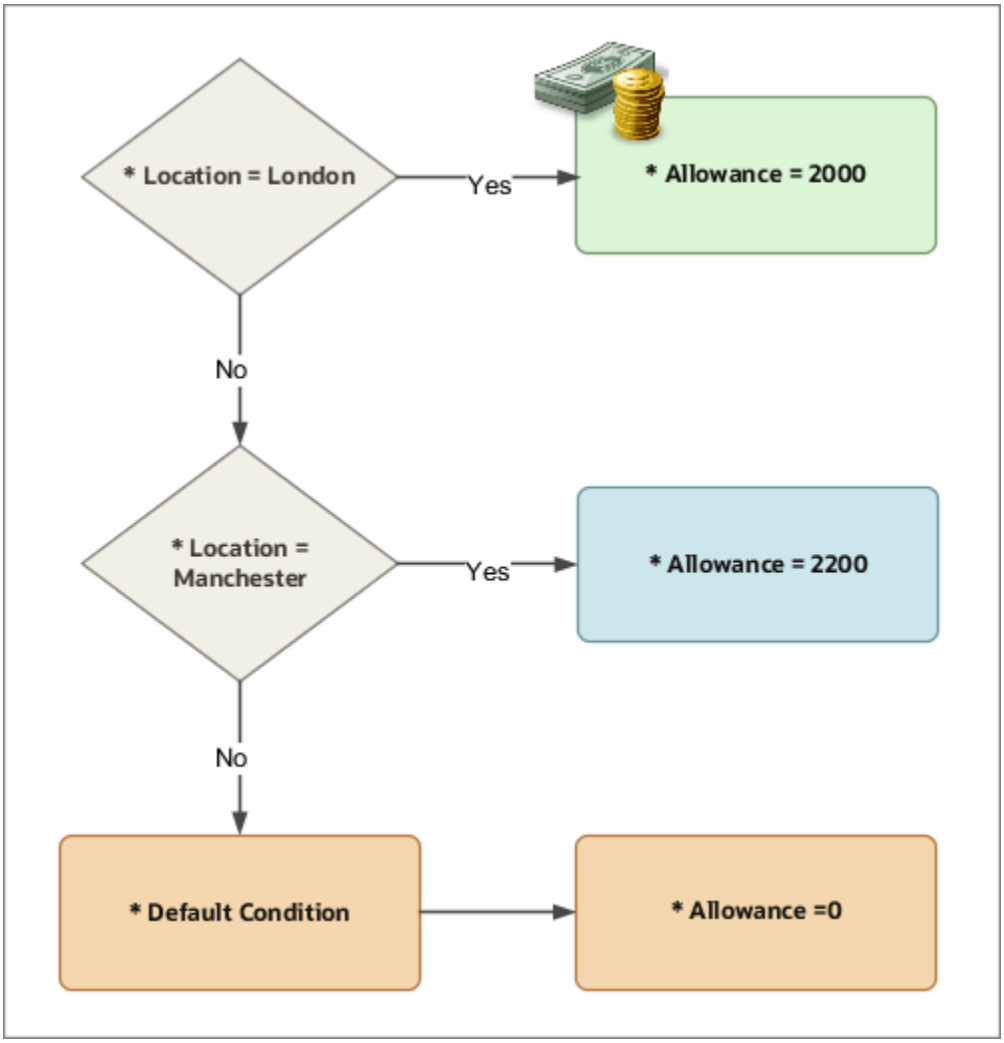
Use HCM Data Loader to define values using one or multiple criteria, such as work location, age, and length of service. You can use values defined by criteria in rate definitions and formula.

Let's consider an example. A weighting allowance is paid for working in certain locations - London 2000, Manchester 2200.

In this case, you will define these value definitions:

- One value definition for the overall weighting allowance.
- One criteria value definition for each location criteria, such as London and Manchester.
- One default criteria value definition to cover all other locations.

You could also define a value definition for each leaf node value.



Before You Begin

Follow these guidelines and considerations:

- When creating a child value definition, ensure that you specify the parent value definition first.
- Before referencing criteria value definitions, ensure that any database items required for the definitions exist.
- Create value sets required to validate criteria value definition values.

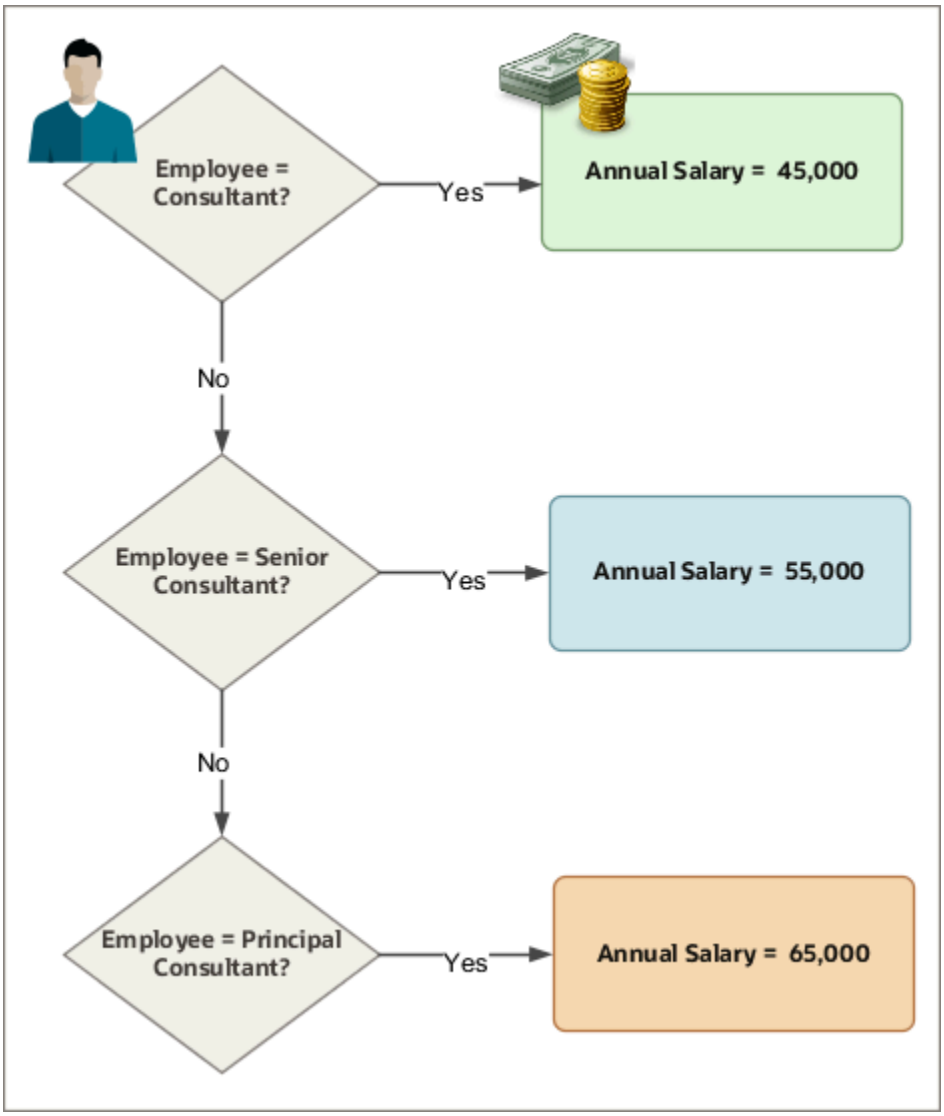
Loading Value Defined By Criteria for Annual Salaries

Use HCM Data Loader to calculate annual salaries for employees based on their position by defining conditions and values for salaries.

In this example and illustration,

- If the employee is a Consultant, pay 45,000.
- If the employee is a Senior Consultant, pay 55,000.

- If the employee is a Principal Consultant, pay 65,000.



This `ValueDefinition.dat` file loads the conditions and salary values.

```
METADATA|ValueDefinition|BaseName|DateMode|DefaultFlag|EffectiveEndDate|EffectiveStartDate|Expression|
Periodicity|Sequence|ValueIdentifier|ValueSetCode1|ValueGroupName|CalculationTypeName|DatabaseItemName|
DefaultCalculationTypeName|ParentValueDefinitionName|LegislativeDataGroupName|ValueDefinitionName|
DisplayTag|CurrencyCode|Uom|RootValueDefinitionName
METADATA|RangeItem|EffectiveStartDate|EffectiveEndDate|LowValue|HighValue|Value1|LowValueText|
LegislativeDataGroupName|ValueDefinitionName
MERGE|ValueDefinition|VBC_Job|E||4712/12/31|2020/01/01||||||Salary|Value By Criteria||Flat Amount||CRFL RRF
 LDG US1|VBC_Job||||VBC_Job
MERGE|ValueDefinition|Job_Consultant|E|N|4712/12/31|2020/01/01|=||1|||Salary|Node|PER_ASG_JOB_NAME||VBC_Job|
CRFL RRF LDG US1|Job_Consultant||||VBC_Job
MERGE|ValueDefinition|Consultant_Salary|E||4712/12/31|2020/01/01||Year||||Salary|Flat Amount|||
Job_Consultant|CRFL RRF LDG US1|Consultant_Salary||USD|M|VBC_Job
MERGE|ValueDefinition|VBC_Senior Consultant|E|N|4712/12/31|2020/01/01|=||2|||Salary|Node|PER_ASG_JOB_NAME||
VBC_Job|CRFL RRF LDG US1|VBC_Senior Consultant||||VBC_Job
MERGE|ValueDefinition|Senior Consultant_Salary|E||4712/12/31|2020/01/01||Year||||Salary|Flat Amount|||
VBC_Senior Consultant|CRFL RRF LDG US1|Senior Consultant_Salary||USD|M|VBC_Job
MERGE|ValueDefinition|VBC_Principal Consultant|E|N|4712/12/31|2020/01/01|=||3|||Salary|Node|
PER_ASG_JOB_NAME||VBC_Job|CRFL RRF LDG US1|VBC_Principal Consultant||||VBC_Job
```

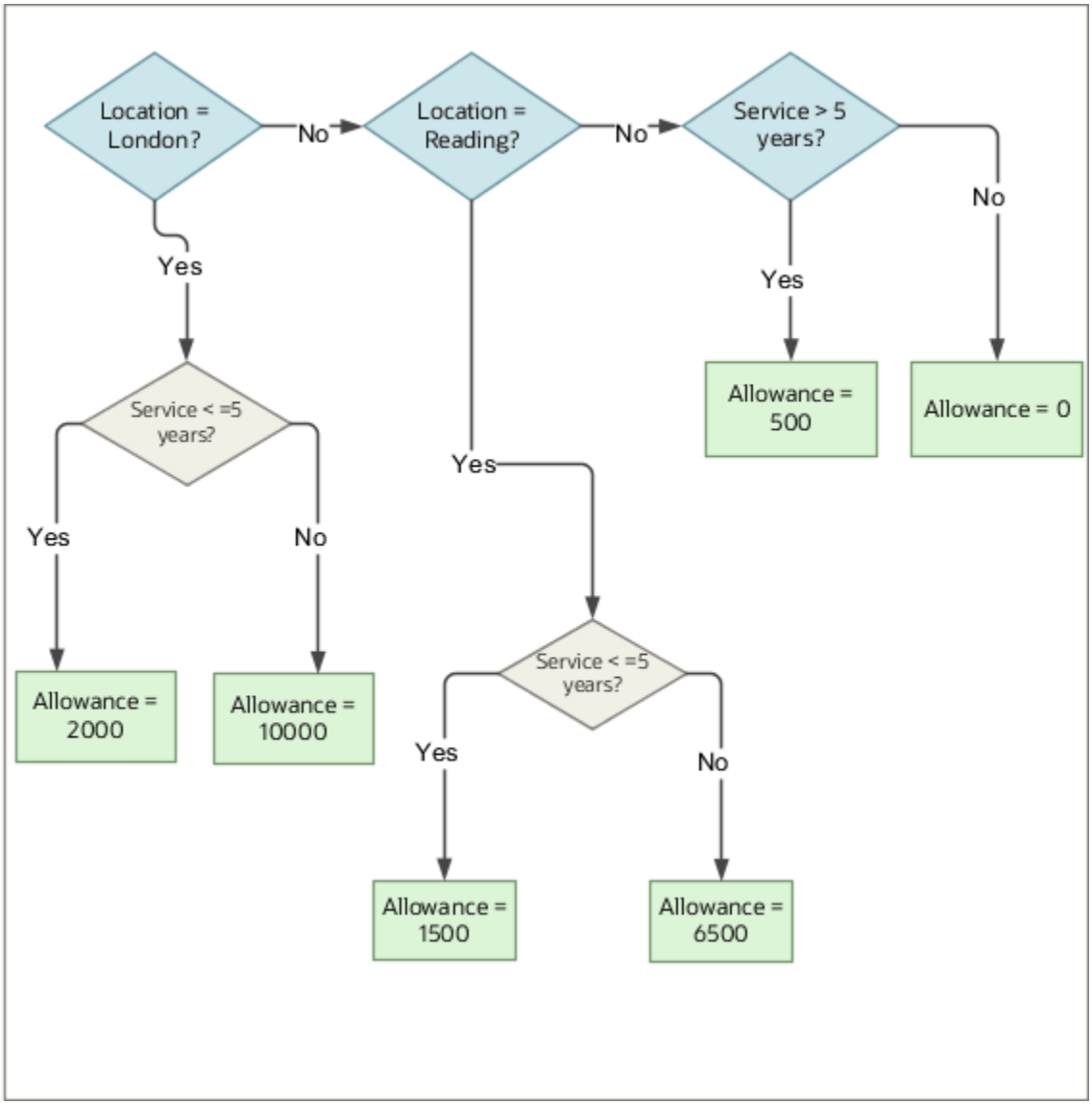
```
MERGE|ValueDefinition|Principal Consultant_Salary|E||4712/12/31|2020/01/01||Year||||Salary|Flat Amount|||
VBC_Principal Consultant|CRFL RRF LDG US1|Principal Consultant_Salary||USD|M|VBC_Job
MERGE|RangeItem|2020/01/01|4712/12/31||||Consultant|CRFL RRF LDG US1|Job_Consultant
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|45000||CRFL RRF LDG US1|Consultant_Salary
MERGE|RangeItem|2020/01/01|4712/12/31||||Senior Consultant|CRFL RRF LDG US1|VBC_Senior Consultant
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|55000||CRFL RRF LDG US1|Senior Consultant_Salary
MERGE|RangeItem|2020/01/01|4712/12/31||||Principal Consultant|CRFL RRF LDG US1|VBC_Principal Consultant
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|65000||CRFL RRF LDG US1|Principal Consultant_Salary
```

Loading Housing Allowance Based On Location

Let's look at an example where we use HCM Data Loader to calculate housing allowance for employees by defining conditions and allowance values.

To pay a housing allowance based on location, you could set up these criteria for an employee working in sales:

- If they have less than or equal to 5 years of service and if the location is London, pay a housing allowance of 2,000.
- If they have greater than 5 years of service and if the location is London, pay a housing allowance of 10,000.
- If they have less than or equal to 5 years of service and if the location is Reading, pay a housing allowance of 1,500.
- If they have greater than 5 years of service and if the location is Reading, pay a housing allowance of 6,500.
- If they have greater than 5 years of service and any other location, pay a housing allowance of 500. This criteria is the default one.



This `ValueDefinition`. dat file loads the conditions and bonus values.

```
MERGE|ValueDefinition|VBC Allowance Location|E||4712/12/31|2020/01/01|||||Allowance|Value By Criteria||Flat
 Amount||CRFL RRF LDG US1|VBC Allowance Location||||VBC Allowance Location
MERGE|ValueDefinition|VBC_Location|E|N|4712/12/31|2020/01/01|=||1|||Allowance|Node|PER_ASG_LOCATION_NAME||
VBC Allowance Location|CRFL RRF LDG US1|VBC_Location||||VBC Allowance Location
MERGE|ValueDefinition|VBC_Service|E|N|4712/12/31|2020/01/01|<=||1|||Allowance|Node|
PER_ASG_LENGTH_OF_SERVICE_BY_SENIORITY||VBC_Location|CRFL RRF LDG US1|VBC_Service||||VBC Allowance Location
MERGE|ValueDefinition|VBC_Allowance|E||4712/12/31|2020/01/01||Year||||Allowance|Flat Amount|||VBC_Service|
CRFL RRF LDG US1|VBC_Allowance||USD|M|VBC Allowance Location
MERGE|ValueDefinition|VBC_Seniority greater than|E|N|4712/12/31|2020/01/01|>||2|||Allowance|Node|
PER_ASG_LENGTH_OF_SERVICE_BY_SENIORITY||VBC_Location|CRFL RRF LDG US1|VBC_Seniority greater than||||VBC
 Allowance Location
MERGE|ValueDefinition|VBC_Location_Reading|E|N|4712/12/31|2020/01/01|=||2|||Allowance|Node|
PER_ASG_LOCATION_NAME||VBC Allowance Location|CRFL RRF LDG US1|VBC_Location_Reading||||VBC Allowance
 Location
MERGE|ValueDefinition|VBC-service lessthanequal 5|E|N|4712/12/31|2020/01/01|<=||1|||Allowance|Node|
PER_ASG_LENGTH_OF_SERVICE_BY_SENIORITY||VBC_Location_Reading|CRFL RRF LDG US1|VBC-service lessthanequal
 5||||VBC Allowance Location
MERGE|ValueDefinition|VBC_Allowance_Reading<=5|E||4712/12/31|2020/01/01||Year||||Allowance|Flat Amount|||
VBC-service lessthanequal 5|CRFL RRF LDG US1|VBC_Allowance_Reading<=5||USD|M|VBC Allowance Location
```

```
MERGE|ValueDefinition|VBC_Allowance_Reading>5|E||4712/12/31|2020/01/01||Year||||Allowance|Flat Amount|||
VBC_allowance_Reading>5|CRFL RRF LDG US1|VBC_Allowance_Reading>5||USD|M|VBC Allowance Location
MERGE|ValueDefinition|VBC_allowance_Reading>5|E|N|4712/12/31|2020/01/01|>||2|||Allowance|Node|
PER_ASG_LENGTH_OF_SERVICE_BY_SENIORITY||VBC_Location_Reading|CRFL RRF LDG US1|VBC_allowance_Reading>5||||VBC
 Allowance Location
MERGE|ValueDefinition|VBC_Value_All_>5|E||4712/12/31|2020/01/01||Year||||Allowance|Flat Amount|||
VBC_Lengthservice>5|CRFL RRF LDG US1|VBC_Value_All_>5||USD|M|VBC Allowance Location
MERGE|ValueDefinition|VBC_Lengthservice>5|E|N|4712/12/31|2020/01/01|>||3|||Allowance|Node|
PER_ASG_LENGTH_OF_SERVICE_BY_SENIORITY||VBC Allowance Location|CRFL RRF LDG US1|VBC_Lengthservice>5||||VBC
 Allowance Location
MERGE|ValueDefinition|VBC_Allowance_greaterthan 5|E||4712/12/31|2020/01/01||Year||||Allowance|Flat Amount|||
VBC_Seniority greater than|CRFL RRF LDG US1|VBC_Allowance_greaterthan 5||USD|M|VBC Allowance Location
MERGE|ValueDefinition|VBC_Default_Criteria|E|Y|4712/12/31|2020/01/01||||||Allowance|Node|||VBC Allowance
 Location|CRFL RRF LDG US1|VBC_Default_Criteria||||VBC Allowance Location
MERGE|ValueDefinition|Def_criter_Val|E||4712/12/31|2020/01/01||Year||||Allowance|Flat Amount|||
VBC_Default_Criteria|CRFL RRF LDG US1|Def_criter_Val||USD|M|VBC Allowance Location
MERGE|RangeItem|2020/01/01|4712/12/31||||London|CRFL RRF LDG US1|VBC_Location
MERGE|RangeItem|2020/01/01|4712/12/31||||5|CRFL RRF LDG US1|VBC_Service|
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|2000||CRFL RRF LDG US1|VBC_Allowance
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|10000||CRFL RRF LDG US1|VBC_Allowance_greaterthan 5
MERGE|RangeItem|2020/01/01|4712/12/31||||5|CRFL RRF LDG US1|VBC-service lessthanequal 5
MERGE|RangeItem|2020/01/01|4712/12/31||||5|CRFL RRF LDG US1|VBC_Seniority greater than
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|1500||CRFL RRF LDG US1|VBC_Allowance_Reading<=5
MERGE|RangeItem|2020/01/01|4712/12/31||||5|CRFL RRF LDG US1|VBC_allowance_Reading>5
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|6500||CRFL RRF LDG US1|VBC_Allowance_Reading>5
MERGE|RangeItem|2020/01/01|4712/12/31||||5|CRFL RRF LDG US1|VBC_Lengthservice>5
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|500||CRFL RRF LDG US1|VBC_Value_All_>5
MERGE|RangeItem|2020/01/01|4712/12/31||||Reading|CRFL RRF LDG US1|VBC_Location_Reading
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|0.00||CRFL RRF LDG US1|Def_criter_Val
```

Loading Allowance Payments Based on Percentage of Salary

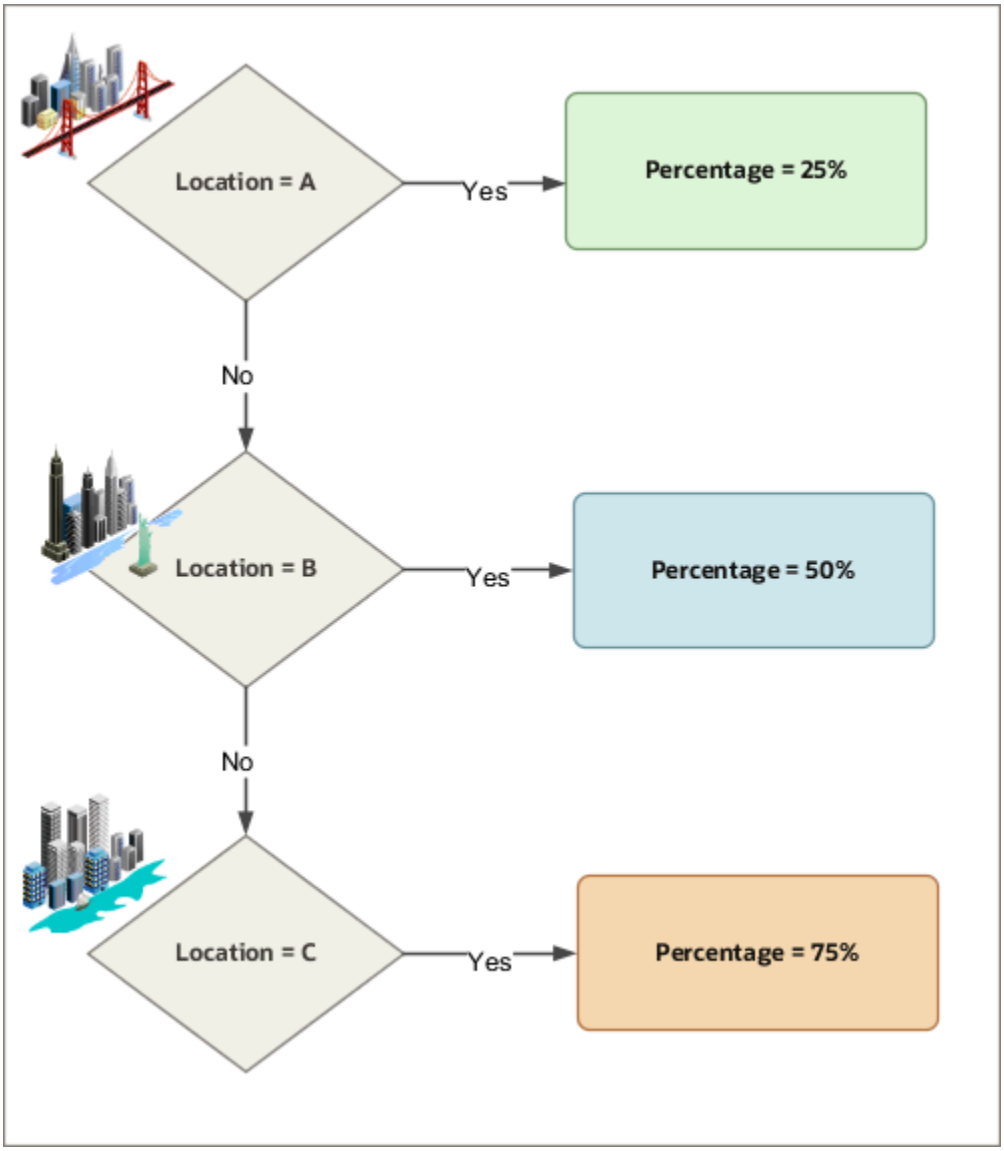
Use HCM Data Loader to calculate allowance payments that are based on a percentage of salary. The percentage rate is based on a condition, such as worker's location.

Here are some criteria that you could use:

- For workers at location A, apply a percentage of 25% (0.25)
- For workers at the location B, apply a percentage of 50% (0.50)
- For workers at the location C, apply a percentage of 75% (0.75)

Note: You must set the default calculation type as Number.

This figure shows the criteria that you can use to calculate the percentage rate.



This `ValueDefinition. dat` file loads the conditions and allowance values.

```
MERGE|ValueDefinition|VBC_Percent_Salary|E||4712/12/31|2020/01/01||||||Salary|Value By Criteria||Number||
CRFL RRF LDG US1|VBC_Percent_Salary||||VBC_Percent_Salary
MERGE|ValueDefinition|VBC_Location A|E|N|4712/12/31|2020/01/01|=||1|||Salary|Node|PER_ASG_LOCATION_NAME||
VBC_Percent_Salary|CRFL RRF LDG US1|VBC_Location A||||VBC_Percent_Salary
MERGE|ValueDefinition|Location A Percentage|E||4712/12/31|2020/01/01||Year||||Salary|Number|||VBC_Location
 A|CRFL RRF LDG US1|Location A Percentage|||N|VBC_Percent_Salary
MERGE|ValueDefinition|VBC_Location B|E|N|4712/12/31|2020/01/01|=||2|||Salary|Node|PER_ASG_LOCATION_NAME||
VBC_Percent_Salary|CRFL RRF LDG US1|VBC_Location B||||VBC_Percent_Salary
MERGE|ValueDefinition|Location B Percentage|E||4712/12/31|2020/01/01||Year||||Salary|Number|||VBC_Location
 B|CRFL RRF LDG US1|Location B Percentage|||N|VBC_Percent_Salary
MERGE|ValueDefinition|VBC_Location C|E|N|4712/12/31|2020/01/01|=||3|||Salary|Node|PER_ASG_LOCATION_NAME||
VBC_Percent_Salary|CRFL RRF LDG US1|VBC_Location C||||VBC_Percent_Salary
MERGE|ValueDefinition|Location C Percentage|E||4712/12/31|2020/01/01||Year||||Salary|Number|||VBC_Location
 C|CRFL RRF LDG US1|Location C Percentage|||N|VBC_Percent_Salary
MERGE|RangeItem|2020/01/01|4712/12/31||||A|CRFL RRF LDG US1|VBC_Location A
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|0.25||CRFL RRF LDG US1|Location A Percentage
MERGE|RangeItem|2020/01/01|4712/12/31||||B|CRFL RRF LDG US1|VBC_Location B
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|0.5||CRFL RRF LDG US1|Location B Percentage
MERGE|RangeItem|2020/01/01|4712/12/31||||C|CRFL RRF LDG US1|VBC_Location C
```

```
MERGE|RangeItem|2020/01/01|4712/12/31|0|999999999999|0.75||CRFL RRF LDG US1|Location C Percentage
```

Event Groups and Event Notification

Overview of Loading Payroll Event Groups

Use HCM Data Loader to load an event group, which is a collection of related events that are monitored for various features.

Types of Payroll Event Groups

As this table shows, you can define these three types of payroll event groups:

Event Group	Use It To
Proration	Calculate proportionate earnings and deduction amounts whenever payroll relevant data changes during a payroll period.
Retroactive	Ensure that your payroll run for the current period reflects any backdated payments and deductions from previous payroll periods.
Event Action	Trigger actions, such as the recalculation of a time card or the creation of payroll actions, for a late new hire. This type of event group can be associated with multiple event actions. For example, an event group that tracks changes to the assignment details of an employee can be associated with these events: • Timecard Resubmission • HCM Rates Recalculation

Event Group Components

This table illustrates the HCM Data Loader components that can be defined for each type of payroll event group.

Component	Description	Proration	Retroactive	Event Action
Event Group	Group of related events, such as changes to an employee's assignment, being monitored for features such as the retroactive recalculation of payroll and time cards, or the generation of HCM rates.	Yes	Yes	Yes
Date Tracked Event	The type of changes monitored by each event group, such as the creation	Yes	Yes	Yes

Component	Description	Proration	Retroactive	Event Action
	of an element entry or updates to an employee's assignment.			
Event Value Changes	Criteria to qualify a change that initiates or disqualifies an event, such as change from job director to senior director.	No	No	Yes
Event Value Qualifier	Additional contextual information to enforce when determining the validity of the event value change. For example, when a position changes from job director to senior director on the primary assignment rather than a secondary assignment. The assignment level is the qualifier in this example. **Note:** For more information on qaulifiers, see *Qualifiers for Event Actions*.	No	No	Yes
Event Group Action	Association between an action and event groups, such as initiating retroactive calculations when a job and position changes.	No	No	Yes

Related Topics

- Guidelines for Loading Time and Labor Event Groups
- Examples of Loading Time and Labor Event Groups
- Overview of Payroll Event Groups

Overview of Payroll Event Entities

This table provides the list of entities supported by payroll events.

Entity	Table	Name
`oracle.apps.hcm.addresses.publicMode`	`PER_ADDRESSES_F`	`AddressDEO`

Entity	Table	Name
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	ApplicantAssignmentDEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	ApplicantTermsDEO
oracle.apps.hcm.payrolls.registratio	PAY_ASSIGNED_PAYROLLS_F	Assigned Payroll
oracle.apps.hcm.employment.core.publi	PER_ASSIGNMENT_SUPERVISORS_F	AssignmentSupervisorDEO
oracle.apps.hcm.employment.core.publi	PER_ASSIGN_WORK_MEASURES_F	AssignmentWorkMeasureDEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	BenefitsAssignmentDEO
oracle.apps.hcm.people.core.protecte	PER_CITIZENSHIPS	CitizenshipEO
oracle.apps.hcm.people.core.protecte	PER_CONTACT_RELSHIPS_F	ContactRelationshipDEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	CWKAssignmentDEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	CWKTermsDEO
oracle.apps.hcm.documentsOfRecord.co	HR_DOCUMENTS_OF_RECORD	DocumentsOfRecordEO
oracle.apps.hcm.payrolls.elements.en	PAY_ELEMENT_ENTRIES_F	Element Entry
oracle.apps.hcm.payrolls.elements.en	PAY_ELEMENT_ENTRY_VALUES_F	Element Entry Value
oracle.apps.hcm.people.core.protecte	PER_EMAIL_ADDRESSES	EmailAddressEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	EmployeeAssignmentDEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	EmployeeTermsDEO
oracle.apps.hcm.people.core.protecte	PER_ETHNICITIES	EthnicityEO
oracle.apps.hcm.workStructures.grade	PER_GRADES_F	GradeDEO
oracle.apps.hcm.workStructures.jobs.m	PER_JOBS_F	JobDEO
oracle.apps.hcm.workStructures.jobs.m	PER_JOB_FAMILY_F	JobFamilyDEO
oracle.apps.hcm.workStructures.jobs.m	PER_JOB_FAMILY_F_TL	JobFamilyTranslationEO

Entity	Table	Name
oracle.apps.hcm.locations.model.enti	PER_LOCATION_DETAILS_F	LocationDetailDEO
oracle.apps.hcm.locations.model.enti	PER_LOCATIONS	LocationEO
oracle.apps.hcm.people.core.protecte	PER_NATIONAL_IDENTIFIERS	NationalIdentifierEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	NWAssignmentDEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	NWTermsDEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	OfferAssignmentDEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	OfferTermsDEO
oracle.apps.hcm.rates.protectedModel	PAY_RATE_VALUES	Pay Rate Values
oracle.apps.hcm.employment.core.publi	PER_PERIODS_OF_SERVICE	PeriodOfServiceEO
oracle.apps.hcm.globalAbsences.recor	ANC_PER_ABS_ENTRIES	PersonAbsEntryEO
oracle.apps.hcm.globalAbsences.recor	ANC_PER_ABS_MATERNITY	PersonAbsMaternityEO
oracle.apps.hcm.people.core.protecte	PER_ALL_PEOPLE_F	PersonDEO
oracle.apps.hcm.people.healthAndSafe	PER_DISABILITIES_F	PersonDisabilityDEO
oracle.apps.hcm.people.core.protecte	PER_PEOPLE_EXTRA_INFO_F	PersonExtraInfoDEO
oracle.apps.hcm.people.core.protecte	PER_PEOPLE_LEGISLATIVE_F	PersonLegislativeInfoDEO
oracle.apps.hcm.people.core.protecte	PER_PERSON_NAMES_F	PersonNameDEO
oracle.apps.hcm.people.core.protecte	PER_PHONES	PhoneEO
oracle.apps.hcm.workStructures.posit	PositionDEO	PositionDEO
oracle.apps.hcm.profiles.core.protec	HRT_PROFILE_ITEMS	ProfileItemEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	PWAssignmentDEO
oracle.apps.hcm.employment.core.publi	PER_ALL_ASSIGNMENTS_M	PWTermsDEO

Entity	Table	Name
oracle.apps.hcm.people.core.protecte	PER_RELIGIONS	ReligionEO
oracle.apps.hcm.compensation.salary.	CMP_SALARY	SalaryEO
oracle.apps.hcm.employment.workSched	PER_SCHEDULE_ASSIGNMENTS	Schedule Assignment Entity
oracle.apps.hcm.employment.workSched	PER_SCHEDULE_EXCEPTIONS	Schedule Exception Entity
oracle.apps.hcm.absenceManagement.wo	PER_SCHEDULE_EXCEPTIONS	Schedule Exception Entity
oracle.apps.hcm.absenceManagement.wo	PER_SCHEDULE_EXCEPTIONS	Schedule Exception Entity
oracle.apps.hcm.employment.core.publi	PER_WORKING_HOUR_PATTERNS_F	WorkingHourPatternDEO
oracle.apps.hcm.payrolls.deductions.	PAY_RANGE_ITEMS_F	RangeItemDEO
oracle.apps.hcm.payrolls.deductions.	PAY_VALUE_DEFINITIONS_F	ValueDefinitionDEO
oracle.apps.hcm.employment.core.publi	PER_PEOPLE_GROUPS	PeopleGroupEO
oracle.apps.hcm.payrolls.registratio	PAY_ASSIGNED_PAYROLLS_DN	PayrollUsageEO
oracle.apps.hcm.people.core.protecte	PER_PERSONS	PersonEO

Related Topics

- Guidelines for Loading Time and Labor Event Groups
- Examples of Loading Time and Labor Event Groups

Example of Loading Retroactive Payroll Calculation Event Groups

Assume you need to monitor retrospective salary and element entry changes so you can calculate the necessary payroll adjustments.

For example, Lucy Hall is awarded a backdated salary increase effective from 1st June. An event is required to identify the salary change that took place on this date. The event enables retropay to calculate the salary arrears that are payable to Lucy.

Create an event group for the retroactive recalculation of payroll for your legislative data group.

This example EventGroup.dat file loads a retropay event group that tracks salary and element entry.

```
METADATA|EventGroup|EventGroupCode|LegislativeDataGroupName|EventGroupName|EventGroupType|ProrationType
```

```
MERGE|EventGroup|EvtGrpCD_Salary|US Legislative Data Group|EvtGrp_Salary|R|
METADATA|DateTrackedEvent|EventGroupCode|LegislativeDataGroupName|ColumnName|UpdateType|ProrationStyle|
DatedObjectName
MERGE|DateTrackedEvent|EvtGrpCD_Salary|US Legislative Data Group|SALARY_AMOUNT|DT_UPDATE_COLUMN||
oracle.apps.hcm.compensation.salary.core.protectedModel.entity.SalaryEO
MERGE|DateTrackedEvent|EvtGrpCD_Salary|US Legislative Data Group|SCREEN_ENTRY_VALUE|DT_CORRECTION||
oracle.apps.hcm.payrolls.elements.entries.protectedModel.entity.ElementEntryValueDEO
MERGE|DateTrackedEvent|EvtGrpCD_Salary|US Legislative Data Group|SCREEN_ENTRY_VALUE|DT_UPDATE_COLUMN||
oracle.apps.hcm.payrolls.elements.entries.protectedModel.entity.ElementEntryValueDEO
MERGE|DateTrackedEvent|EvtGrpCD_Salary|US Legislative Data Group||DT_INSERT||
oracle.apps.hcm.compensation.salary.core.protectedModel.entity.SalaryEO
MERGE|DateTrackedEvent|EvtGrpCD_Salary|US Legislative Data Group||DT_INSERT||
oracle.apps.hcm.payrolls.elements.entries.protectedModel.entity.ElementEntryValueDEO
```

Note: You can use the predefined event groups for retroactively recalculating payroll and for prorating the earnings and deductions.

Related Topics

- Guidelines for Loading Time and Labor Event Groups
- Examples of Loading Time and Labor Event Groups

Example of Loading Retropay for Late Hires Event Groups

Let's consider an example of a late hire employee. The event group identifies the late hired employee.

To enable retropay for late hires, create a new Event Group for their legislative data group.

Note: Even if you are currently using the Late Hire functionality, you don't need to create a new Event Group.

Use this reference data as template to update your new event group:

```
METADATA|EventGroup|EventGroupCode|EventGroupName|EventGroupType|LegislativeDataGroupName
METADATA|DateTrackedEvent|ColumnName|UpdateType|EventGroupCode|DatedObjectName|LegislativeDataGroupName
MERGE|EventGroup|<Event_Group_Code>|<Event_Group_Name>|A|<Legislative Data Group Name>
MERGE|DateTrackedEvent||DT_INSERT|<Event_Group_Code>|
oracle.apps.hcm.payrolls.registrations.protectedModel.entity.PayrollUsageDEO|<Legislative Data Group Name>
```

Update your Event Group in the reference data using these details:

- Replace <Event_Group_Code> with event group code, such as MY_EVENT_GROUP
- Replace <Event_Group_Name> with event group name, such as My_Event_Group
- Replace <Legislative Data Group Name> with LDG Name, such as US Legislative Data group
- Zip the modified template as EventGroup.zip.

Related Topics

- Guidelines for Loading Time and Labor Event Groups
- Examples of Loading Time and Labor Event Groups

Example of Loading Retropay for Late Hires - Move Hire Date

An employee is entered in the application with an incorrect hire date of 01-July-2021. Subsequently, you correct the hire date to 01-June-2021 and now the employee is eligible for payroll payments for the June payroll period.

These example EventGroup and EventAction dat files load retropay for late hire event actions that involve a change in the hire date.

```
METADATA|EventGroup|EventGroupCode|LegislativeDataGroupName|EventGroupName|EventGroupType|ProrationType
MERGE|EventGroup|EvtGrpCD_MoveDate|US Legislative Data Group|EvtGrp_MoveDate|A|
METADATA|DateTrackedEvent|EventGroupCode|LegislativeDataGroupName|ColumnName|UpdateType|DatedObjectName|
ProrationStyle
MERGE|DateTrackedEvent|EvtGrpCD_MoveDate|US Legislative Data Group||DT_START_EARLIER|
oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignmentDEO|
MERGE|DateTrackedEvent|EvtGrpCD_MoveDate|US Legislative Data Group||DT_START_LATER|
oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignmentDEO|
METADATA|EventGroupAction|EventGroupCode|EventActionCode|EffectiveStartDate
MERGE|EventGroupAction|EvtGrpCD_MoveDate|EvtAct_MoveDate|2016/01/01

METADATA|EventAction|EventActionCode|EventActionName|EffectiveStartDate|ActionSubmission|
EventActionTypeCode|LegislativeDataGroupName
MERGE|EventAction|MY_EVENT_ACTION|My_Event_Action|2016/01/01|SYNC|ORA_CREATE_RUN_REL_ACTIONS|US Legislative
 Data Group
```

Related Topics

- Guidelines for Loading Time and Labor Event Groups
- How Retroactive Pay Is Calculated

Example of Loading Retropay for Late Hires Event Action

The event action creates a payroll action for the late hire, causing the employee to appear in the process results for that payroll with the action status as Retroactively Added.

Calculate retroactive pay for late hires using the enhanced RetroPay Late Hire solution. This feature automates the calculation of retroactive earnings for late hires by removing the need for manual intervention. Also, you can view the late hire details for an employee using the payroll flow and checklist. You can now configure the event action to enable the Evaluate Late Hires flow pattern to:

- Evaluate late hires automatically
- Create a net pay result
- Create retro event notifications

This lets late hires to be processed within the Recalculate Payroll for Retroactive Changes process. This flow pattern contains two tasks:

- Calculate Payroll for Late Hire
- Late Hire Retropay Notification

These tasks replace the need to manually enter an element entry against the late hire, as the Late Hire Retropay Notification task automatically creates the notification. You can now view the late hire details for an employee using the Person Results and Payroll Checklist pages.

Use the sample HCM Data Loader (.DAT file) to enable all or some of these events:

- **Create Flow:** If set to 'Y', generates the Evaluate Late Hire flow pattern with its Calculate Payroll for Late Hire and Late Hire Retropay Notification tasks.
- **Hire Date Later:** If set to 'Y', the retroactive net pay result is generated when an employee's hire date is later than the payroll process date.
- **Run Payroll Relationship Group:** If set to 'Y', the retroactive net pay result is generated even when the original payroll was run with a payroll relationship group.
- **Additional Assignments:** If set to 'Y', the retroactive net pay result is generated when a late assignment is added to a payroll relationship.

Important pages:

- Use the View Flows, Process Results and Person Results pages to view the flow results.
- Use the View Flows page to drill down to the Evaluate Late Hires details.
- Use the Person Results page to drill down to the Calculate Payroll for Late Hires details.

Perform the following steps to enable the flow:

1. Create a new Event Group for your Legislative data group. Follow the steps mentioned in the topic *Example of Loading Retropay for Late Hires Event Groups*
2. Create a new Event Action for your Legislative data group.

   > **Note:** Even if you are currently using the Late Hire functionality, you still need to create a new Event Action.

   Use this reference data as template to update your new event action:

   ```
   METADATA|EventAction|EventActionCode|EffectiveStartDate|EffectiveEndDate|EventActionName|
   EventActionTypeCode|ActionSubmission|ProcessMode|AutoApprove|Description|LegislativeDataGroupName
   METADATA|EventGroupAction|EventActionCode|EffectiveStartDate|EffectiveEndDate|EventGroupCode
   MERGE|EventAction|<Event_Action_Code>|1950/01/01|4712/12/31|<Event_Action_Name>|
   ORA_CREATE_RUN_REL_ACTIONS|SYNC|AUTO|N|<Event_Action_Description>|<Legislative Data Group Name>
   MERGE|EventGroupAction|<Event_Action_Code>|1950/01/01|4712/12/31|<Event_Group_Code>
   ```

   Update Event Action in the reference data with following details:

   - Replace <Event_Action_Code> with event action code, such as MY_EVENT_ACTION
   - Replace <Event_Action_Name> with event action name, such as My_Event_Action
   - Replace <Event_Action_Description> with event action description, such as My Event Action
   - Replace <Legislative Data Group Name> with LDG Name, such as US Legislative Data group
   - Replace <Event_Group_Code> with event group code, such as the above example of MY_EVENT_GROUP
   - Zip the modified template as EventAction.zip.
3. Upload EventGroup.zip and EventAction.zip using HDL.
4. Create Event Action HDL files to set the Evaluate Late Hires flow pattern functionality using these samples:

   Sample 1: Only enable flow, Y|N|N|N

   ```
   METADATA|EventAction|FLEX:PAY_EVENT_ACTIONS_DDF|
   CreateFlow(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
   HireDateLater(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
   RunPayRelationshipGroup(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
   AdditionalAssignments(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|EventActionCode|
   ```

```
EffectiveStartDate|EffectiveEndDate|EventActionName|EventActionTypeCode|LookbackTimeDefinitionCode|
ActionSubmission|ProcessMode|AutoApprove|LegislativeDataGroupId
MERGE|EventAction|ORA_CREATE_RUN_REL_ACTIONS|Y|N|N|N|RetroLateHire|1950/01/01|4712/12/31|RetroLateHire|
ORA_CREATE_RUN_REL_ACTIONS||SYNC|MANUAL|Y|300100003140635
```

Sample 2: Enable flow with Hire Date Later, Y|Y|N|N

```
METADATA|EventAction|FLEX:PAY_EVENT_ACTIONS_DDF|
CreateFlow(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
HireDateLater(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
RunPayRelationshipGroup(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
AdditionalAssignments(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|EventActionCode|
EffectiveStartDate|EffectiveEndDate|EventActionName|EventActionTypeCode|LookbackTimeDefinitionCode|
ActionSubmission|ProcessMode|AutoApprove|LegislativeDataGroupId
MERGE|EventAction|ORA_CREATE_RUN_REL_ACTIONS|Y|Y|N|N|RetroLateHire|1950/01/01|4712/12/31|RetroLateHire|
ORA_CREATE_RUN_REL_ACTIONS||SYNC|MANUAL|Y|300100003140635
```

Sample 3: Enable flow with Run Payroll Relationship Group, Y|N|Y|N

```
METADATA|EventAction|FLEX:PAY_EVENT_ACTIONS_DDF|
CreateFlow(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
HireDateLater(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
RunPayRelationshipGroup(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
AdditionalAssignments(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|EventActionCode|
EffectiveStartDate|EffectiveEndDate|EventActionName|EventActionTypeCode|LookbackTimeDefinitionCode|
ActionSubmission|ProcessMode|AutoApprove|LegislativeDataGroupId
MERGE|EventAction|ORA_CREATE_RUN_REL_ACTIONS|Y|N|Y|N|RetroLateHire|1950/01/01|4712/12/31|RetroLateHire|
ORA_CREATE_RUN_REL_ACTIONS||SYNC|MANUAL|Y|300100003140635
```

Sample 4: Enable flow with Additional Assignments, Y|N|N|Y

```
METADATA|EventAction|FLEX:PAY_EVENT_ACTIONS_DDF|
CreateFlow(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
HireDateLater(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
RunPayRelationshipGroup(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|
AdditionalAssignments(PAY_EVENT_ACTIONS_DDF=ORA_CREATE_RUN_REL_ACTIONS)|EventActionCode|
EffectiveStartDate|EffectiveEndDate|EventActionName|EventActionTypeCode|LookbackTimeDefinitionCode|
ActionSubmission|ProcessMode|AutoApprove|LegislativeDataGroupId
MERGE|EventAction|ORA_CREATE_RUN_REL_ACTIONS|Y|N|N|Y|RetroLateHire|1950/01/01|4712/12/31|RetroLateHire|
ORA_CREATE_RUN_REL_ACTIONS||SYNC|MANUAL|Y|300100003140635
```

Zip the modified template as EvaluateLateHire.zip.

5. Upload EvaluateLateHire.zip using HDL.

Note: Original payroll must have a run type of **Regular**.

Related Topics

- Guidelines for Loading Time and Labor Event Groups

Example of loading Event Notifications

This example describes how you can load notifications for retroactive events.

Let's consider this scenario:

In this example, Vision Corp organization owns a chain of supermarkets. Few employees are eligible to receive **Premium Allowance** payment when they work in a different job within the supermarket. For example, when an employee who normally works as a butcher, works at the baking counter. Vision Corp maintains the **Premium Allowance** payment rates on a user-defined table.

The **Premium Allowance** rates are increased effective September 1st but Vision Corp doesn't enter the new rates on the user-defined table until November. As a result, any employee who has received **Premium Allowance** payments in September and October are eligible to be paid arrears.

The Payroll Manager uses HCM Data Loader to create retroactive event notifications for all impacted employees. The process date of the event notifications is set to September 1st.

```
COMMENT Data for Business Object: PayrollEventNotification Version: V1 Created on: 09-12-2021
COMMENT New retro notifications to support Premium Allowance rates increase as of 1-Sep-2021
METADATA|PayrollEventNotification|ApprovalStatus|EventReportType|ProcessDate|AssignmentNumber|OverrideDate|
EventActionCode
MERGE|PayrollEventNotification|A|RETRONOT|2021/09/01|E000012
MERGE|PayrollEventNotification|A|RETRONOT|2021/09/01|E000013
MERGE|PayrollEventNotification|A|RETRONOT|2021/09/01|E000014
MERGE|PayrollEventNotification|A|RETRONOT|2021/09/01|E000015
MERGE|PayrollEventNotification|A|RETRONOT|2021/09/01|E000016
```

Example of Entering an Override Date for a Retro Event Notification

This example describes how you can enter an override date for a retroactive event notification.

Let's consider this scenario: The Payroll Manager has manually adjusted an employee's earnings and tax calculations for the July 30th payroll period. A retroactive event notification exists for the employee with a process date of July 5th, but the Payroll Manager wants to stop retropay processing arrears for the employee for the July period.

Using HCM Data Loader, the Payroll Manager enters an Override Date on the retroactive event notification of August 1st for the employee. This ensures that retropay calculations are performed from August 1st for this retroactive notification.

```
COMMENT Data for Business Object: PayrollEventNotification Version: V1 Created on: 09-12-2021
METADATA|PayrollEventNotification|ApprovalStatus|EventReportType|ProcessDate|AssignmentNumber|OverrideDate
MERGE|PayrollEventNotification|A|RETRONOT|2021/07/05|E000017|2021/08/01
MERGE|PayrollEventNotification|A|RETRONOT|2021/07/05|E000018|2021/08/01
MERGE|PayrollEventNotification|A|RETRONOT|2021/07/05|E000019|2021/08/01
MERGE|PayrollEventNotification|A|RETRONOT|2021/07/05|E000020|2021/08/01
MERGE|PayrollEventNotification|A|RETRONOT|2021/07/05|E000021|2021/08/01
```

Overview of Loading Event Actions and Event Action Groups

Use HCM Data Loader to load an event action. An event action links a process to the events within an event group.

Event actions are used by multiple features across HCM, such as retroactive timecard processing and the Generation of HCM rates process. When an event occurs, the process will be triggered based on a set of conditions defined on the event action.

There are different types of event actions, each with its own set of submission control rules. These action codes and names are user enterable free text fields. Here are the valid values for Event Action Type Code with their corresponding meaning:

Event Action Type Code	Meaning
ORA_CREATE_RUN_REL_ACTIONS	Create Historical Run Relationship Actions
ORA_ANC_ACTION_TYPE_RESUBMIT	Absence Resubmission
ORA_HWM_ACTION_TYPE_TC_RESUBMIT	Timecard Resubmission
ORA_HCM_RATES_RECALC	HCM Rates Recalculation
ORA_HRX_MX_SDI_UPDATE	Mexico SDI Update

Submission Types

The process associated to the event action can be run automatically each time an event is triggered for an employee. Alternatively, you can manually submit the process for all employees who have had an event triggered (but not yet processed). There are two event action submission types:

- **Automatic (SYNC):** The application automatically submits processes such as retropay or late hires, each time an event action notification is generated.
- **Manual (MESS):** The application processes the event action notifications, such as Generate HCM Rates notifications, using the 'Process Event Action' flow. The Process Event Action' flow will generate a report that lists all the employees included in the process and any warning or error messages. When using this submission type for the HCM Rates process a report will also be generated which provides details the rates calculated by the process.

Steps to Enable the Event Group and Event Action Feature

1. Create event action(s) for each feature such as to generate HCM rates.
2. Create an event group to identify the type of changes to be monitored such as assignment updates. The event group should have an action type.
3. Associate the event action(s) to the event group to indicate which features use the event group. For example, an event group can be used to monitor changes for retroactive timecard processing and the generation of HCM rates.

Related Topics

- Submit the Process Event Actions Manual Process Flow

Overview of Loading HCM Rates Event Action Groups

The event solution limits the number of employees that have rates re-calculated and improves the performance of the rates calculation process.

Use HCM Data Loader to create an event action and associated event action group for the Generate HCM Rates process.

HCM Rates Submission Type

You can set the submission type of the event action to 'Manual' which indicates the rates will be recalculated when the user submits the 'Process Event Action' flow. This flow will generate reports to validate the results of the rates process:

- **Person Process Status Report:** This report lists all the employees included in the process, the status of their rate calculation, and any warning or error messages.
- **Rates Report:** This report indicates rate values calculated for each employee included in the process, the effective start date of the rate and details of when the rate was last recalculated.

> **Note:** When you implement rates, run the Generate HCM Rates process to calculate and capture rate values for all employees. Going forward, use the Process Event Action flow to recalculate rates for any employee who has an event such as a salary update. It is recommended you run the Process Event Action flow daily.

You also have the option to automatically submit the rates process each time an event notification is raised for an employee, but no reports are generated for this type of action submission.

Enable Payroll Events on Background Flow

If you are monitoring element entry changes in your rates event action group you must enable event processing using the background flow. Set the action parameter 'Process Payroll Events on Background Payroll Flow' to 'Y'.

> **Note:** This step is essential if you are monitoring element entry changes in your rates event action group. This setting will move all payroll event processing to a payroll process which continuously runs in the background. The evaluation and generation of all payroll event notifications and actions will no longer be performed as part of the main transactions such as a new hire.

Rate Events and Payroll Balances

If you calculate rates based on payroll balances you may still need to run the Generate HCM Rates process because changes to this type of rate are not event based.

Event Action Flexfields

HDL supports the flex fields relevant to the event action. For example, if a user is creating event actions for HCM Rates, the rates flex labels will be displayed on the HDL file. When you create an Event Action for HCM Rates Recalculation, specify the flex attributes such as Process Mode to indicate how you want to run the rates process.

The flex attributes for the rate event actions are listed below:

Flex Name	Type	Importance	Values	Flex Description
Process Mode	LOV	Mandatory	Fast	Indicates how the user wants to run the rates process. The method you select will depend on whether you are using the rate for company reporting or a payroll calculation.
				The option 'Fast' is quickest but the least accurate

Flex Name	Type	Importance	Values	Flex Description
				method to calculate rate values. If the rate value is the same on the start date and end date of the processing period it determines the rate value is the same over the entire period. If the start and end values are different, this method then determines where the rate change has occurred.
			Full (Default)	The option 'Full' calculates the rate every day and therefore is the slowest but most accurate method. Value Set: ORA_PAY_RATES_BATCH_PROC_MODE
			Periodic	The option 'Periodic' works in the same manner as 'Fast' except you can specify the duration of each rate calculation period to increase the accuracy of the rate calculation. For example, you can specify a 7-day period to ensure a rate is calculated and compared every 7 days.
Number of Days	LOV	Conditional	Numeric - free text	Indicates the duration of every rate calculation period when using the periodic process mode. Value Set: RATE_INTERVAL

Qualifiers for Event Actions

This feature lets you to specify the type of element entry and HCM rate changes you want to monitor and get notified about.

Define the qualifying conditions that must be met to generate an event action notification. For example, define an event group that only generates event notifications for Salary element entry changes. This feature is available for features that use the event action model, such as HCM Rates.

Qualifier Attributes

When you define qualifiers for an event group, you need to specify the following components for the **EventGroup.dat** file in HCM Data Loader (HDL). The event group must also be associated to an event action.

Name	Description	Required?
Date Track Events	Identifies the entity to be monitored. For example, Element Entry.	Mandatory
Event Value Changes	Identifies the change values to be monitored. You must define these as 'include' or 'exclude' rules in HDL. For example, a US customer can indicate they only want to 'include' input value changes from California to New York. When you don't have a requirement to monitor a specific value change, you should enter **`include <ANY_VALUE> to <ANY_VALUE>`**. **Note:** You can't enter a mix of include and exclude rules for an entity. When you create an 'include' record, the application automatically creates an 'exclude' record (for all other state input value changes), so that you don't need to enter it explicitly. Currently, you must enter the ID of the value. For example, when tracking assignment location changes, you may opt to only track changes from New York to San Francisco. In this instance, you must enter the ID for the New York and San Francisco locations.	Mandatory
Event Value Qualifier	Enables you to qualify the changes to be monitored for the entity. For example, you can indicate you want to monitor the creation of 'Salary' element entries only. You can define multiple qualifiers. For example, you can indicate you want to monitor input value changes to the 'Premium' and 'Shift' element entries only.	Optional

Date Track Actions

When creating records in HDL, you must specify the type of date-track actions to be monitored:

- Insert
- Update
- Correct
- Delete

Note: For event changes, 'insert' and 'delete' aren't supported when you are tracking value changes such as assignment location changes from New York to California. Such a change must always be performed as an update or correction to the assignment record.

Qualifiers

Here's a list of the event qualifiers that are currently supported:

Qualifier	Description
PAY_ELEMENT_ENTRIES ELEMENT_NAME	Qualification of element entry events based on the name of the element. For example, only raise an event notification for 'Salary' element entry changes such as the creation of a salary entry for an employee. **Note:** This doesn't include changes to input values on the element entry.
PAY_ELEMENT_ENTRY_VALUES ELEMENT_NAME	Qualification of element entry input value events based on the name of the element. For example, only raise an event notification when there is a 'Salary' element entry input value change.
PAY_ELEMENT_ENTRY_VALUES INPUT_NAME	Qualification of element entry value events based on the name of the input value. For example, only raise an event notification when there is an 'Amount' input value change on any element entry.
PAY_RATE_VALUES RATE_NAME	Qualification based on the rate amount held on the HCM Rates table. For example, when you run the Generate HCM Rate process and the 'Salary' rate amount is updated for an employee.

These primary flag qualifiers are also supported. However, these aren't intended for Payroll customers:

- EmployeeAssignmentDEO Primary Flag
- CWKTermsDEO Primary Flag
- ApplicantAssignmentDEO Primary Flag
- ApplicantTermsDEO Primary Flag
- BenefitsAssignmentDEO Primary Flag
- CWKAssignmentDEO Primary Flag
- PWTermsDEO Primary Flag
- EmployeeTermsDEO Primary Flag
- NWAssignmentDEO Primary Flag
- NWTermsDEO Primary Flag
- OfferAssignmentDEO Primary Flag
- OfferTermsDEO Primary Flag
- PWAssignmentDEO Primary Flag

Example of Loading Qualifiers for Event Actions

This topic shows how to add qualifiers to an event group using HCM Data Loader (HDL).

Loading Date Tracked Events

The Date Tracked Event component identifies the entity to be monitored. For example, Element Entry.

```
METADATA|DateTrackedEvent|ColumnName|UpdateType|EventGroupCode|DatedObjectName
MERGE|DateTrackedEvent|SCREEN_ENTRY_VALUE|DT_UPDATE_COLUMN|EleEntryChangeEventGroup|
oracle.apps.hcm.payrolls.elements.entries.protectedModel.entity.ElementEntryValueDEO
```

Loading Event Value Changes

The Event Value Change component identifies the change values to be monitored. You must define these as 'include' or 'exclude' rules in HDL. For example, a US customer can indicate they only want to 'include' input value changes from California to New York.

When you don't have a requirement to monitor a specific value change, you should enter `include <ANY_VALUE> to <ANY_VALUE>`.

Note: You can't enter a mix of include and exclude rules for an entity.

When you create an 'include' record, the application automatically creates an 'exclude' record (for all other state input value changes), so that you don't need to enter it explicitly.

Currently, you must enter the ID of the value. For example, when tracking assignment location changes, you may opt to only track changes from New York to San Francisco. In this instance, you must enter the ID for the New York and San Francisco locations.

```
METADATA|EventValueChange|EffectiveStartDate|EffectiveEndDate|Sequence|ValidEvent|FromValue|ToValue|
EventGroupCode|LegislativeDataGroupName|DatedObjectName|ColumnName|UpdateType|ParentEvtValChangeSequence
MERGE|EventValueChange|2020/01/01|4712/12/31|1|Y|<ANY_VALUE>|<ANY_VALUE>|DN_EV_C1|
<LEGISLATIVE_DATA_GROUP_NAME>|oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignmentDEO|
LOCATION_ID|DT_UPDATE_COLUMN|
```

Loading Event Value Qualifiers

The Event Value Qualifier component enables you to qualify the changes to be monitored for the entity. For example, you can indicate you want to monitor the creation of 'Salary' element entries only.

You can define multiple qualifiers. For example, you can indicate you want to monitor input value changes to the 'Premium' and 'Shift' element entries only.

```
METADATA|EventValueQualifier|EffectiveStartDate|EffectiveEndDate|QualifierValue|LegislativeDataGroupName|
ColumnName|UpdateType|EventGroupCode|DatedObjectName|Sequence|QualifierName
MERGE|EventValueQualifier|2020/01/01|4712/12/31|Y|<LEGISLATIVE_DATA_GROUP_NAME>|LOCATION_ID|
DT_UPDATE_COLUMN|DN_EV_C1|oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignmentDEO|1|
EmployeeAssignmentDEO Primary Flag
```

Example of Loading Event Actions for HCM Rates Recalculation

An event action defines how to react to an event, for example, whether to calculate rates when a worker's job or element entries change.

This topic shows how to create an event action using HCM Data Loader. The following rules apply:

- Enable flex information for the generate rates event action by setting FLEX:PAY_EVENT_ACTIONS_DDF to ORA_HCM_RATES_RECALC
- Set the ProcessingType to PAY_EVENT_ACTIONS_DDF=ORA_HCM_RATES_RECALC to indicate this as a rates action type.
- Set the ProcessingType to ORA_FULL to indicate the generate rates process should be run in the 'Full' mode. You also have the option to the run the generate rates process in 'Fast' and 'Periodic' modes.
- It is recommended the ActionSubmission is set to MESS.

  > **Note:** This indicates the rates process will be submitted manually using the new 'Process Event Action' flow. This flow will submit the rates process for all employees who have an unprocessed rates event notification. You can validate the rate results using the reports generated by the 'Process Event Action'. You also have the option to automatically submit the rates process each time an event notification is raised for an employee, but no reports are generated for this type of action submission.

- Provide a name for your event action using the EventActionName attribute.
- Set EventActionTypeCode to ORA_HCM_RATES_RECALC
- Provide a code for your event action using the EventActionCode attribute.
- Set the InformationCategory to ORA_HCM_RATES_RECALC

Let's consider an example where you want to create an event action to submit the HCM Rates Recalculation process. This example EventAction.dat file creates a single Event Action component, which is identified by its user key.

```
METADATA|EventAction|FLEX:PAY_EVENT_ACTIONS_DDF|ProcessingType(PAY_EVENT_ACTIONS_DDF=ORA_HCM_RATES_RECALC)|
ActionSubmission|EffectiveStartDate|EventActionName|EventActionTypeCode|EventActionCode|
LegislativeDataGroupName|InformationCategory
MERGE|EventAction|ORA_HCM_RATES_RECALC|ORA_FULL|MESS|1951/01/01|<CUSTOMER EVENT ACTION NAME>|
ORA_HCM_RATES_RECALC|<CUSTOMER EVENT ACTION CODE>||ORA_HCM_RATES_RECALC
```

Example of Loading Event Group and Associating to an Event Action for HCM Rates Recalculation

This topic shows how to create an event group and associate it to an event action using HCM Data Loader.

Use event groups to identify the type of changes that require the Generate HCM Rates process to be submitted for an employee.

Let's consider an example where you want to track location and element entry changes for an employee. When an event in your group is triggered, a rate event notification will be created to ensure the employee is included in the Generate HCM Rates process.

This example EventGroup.dat file creates a single Event Group component and associates it to the Event Action. For Generate HCM Rates calculations, the EventGroupType attribute value must be A.

Create Event Group

```
METADATA|EventGroup|EventGroupCode|EventGroupName|EventGroupType|LegislativeDataGroupName
MERGE|EventGroup|<CUSTOMER EVENT GROUP CODE>|<CUSTOMER EVENT GROUP NAME>|A|<LDG NAME>
```

Indicate the type of events within the Event Group

```
METADATA|DateTrackedEvent|ColumnName|UpdateType|EventGroupCode|LegislativeDataGroupName|DatedObjectName
MERGE|DateTrackedEvent|LOCATION_ID|DT_UPDATE_COLUMN|GBK_EvtGrp_HCMRate_Fast|ZHRX_US_COST_LDG|
oracle.apps.hcm.employment.core.publicModel.entity.EmployeeAssignmentDEO
MERGE|DateTrackedEvent|SCREEN_ENTRY_VALUE|DT_UPDATE_COLUMN|GK_EvtGrp_HCMRate_EE|CRFL RRF LDG US1|
oracle.apps.hcm.payrolls.elements.entries.protectedModel.entity.ElementEntryValueDEO
```

Associate your Event Action to the Event Group

```
METADATA|EventGroupAction|EffectiveStartDate|EventGroupCode|EventActionCode
MERGE|EventGroupAction|1951/01/01|<CUSTOMER EVENT GROUP CODE>|<CUSTOMER EVENT ACTION CODE
```

16 Loading Payroll Transactional Data

Rates Based on Grades Details

You can calculate rates based on employee's grade details. For example, you can calculate the annual salary rate based on the grade ladder, grade, and grade step information defined for an employee.

When you define a rate definition, you can select a grade rate as a basis for the rate calculation and specify the employment level. The default employment level is Payroll Assignment. The application retrieves the grade ladder details based on assigned grade that's held on an employee's assignment record.

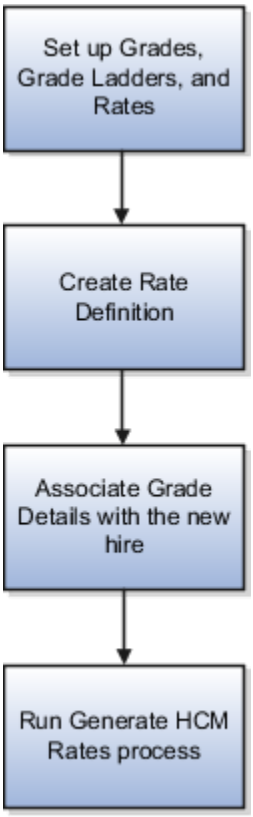
In your rate definition, you add either a rate contributor of type Grade Ladder or Retained Grade.

Calculate Rates Based on Grades

Follow these steps to calculate rates based on grades.

1. Set up grades, grade ladder, and rates for the given job to record the level of compensation for the employees.
2. Create a rate definition based on grade details. Add a rate contributor of type **Grade Ladder**.
3. Hire the employee and provide the required grade and step details.
4. Run the **Generate HCM Rates** process to calculate the rates based on the information held in the grade tables.

This flowchart describes the overall steps to calculate rates based on grade details.



Related Topics

- Example to Calculate Rates Based on Grades With Steps
- Example to Calculate Rates Based on Grades Without Steps
- Example to Calculate Rates Based on Grade Ladder With Multiple Assignments

Example of Loading Rate Definitions for Basic Salary

Use HCM Data Loader to create a primary rate for a basic salary. The salary rate for an employee is displayed on the salary page or used for other purposes, such as a contributor to an absence rate.

Before You Begin

Use HCM Data Loader to create the salary element at the assignment level. The salary element entries contain the information to be retrieved by the salary rate definition for an employee. You can create it using the flat amount or factor calculation rule.

Use this RateDefinitionV2.dat file to create a rate definition and load these values:

Attribute	Value
Category	Element
StorageType	Amount
ElementCode	Regular Salary
ReturnRatePeriodicity	Annually
ConversionFormulaCode	Standard Rate Annualized
ReturnRateCurrency	USD
LegislativeDataGroupName	CRFL RRF LDG US1

```
METADATA|RateDefinitionV2|RateDefinitionCode|ConversionFormulaCode|ContributorTotalPeriodicity|
ReturnRatePeriodicity|EffectiveStartDate|LegislativeDataGroupName|ShortName|Category|SourceSystemId|
SourceSystemOwner|ReturnRateCurrency|ElementCode|Status|BaseSalaryFlag|ReturnDecimalDisplay|
ReturnRoundingRule|StorageType|OverrideAllowed|ReturnFullTimeRateFlag|ReportingRateFlag|
CalculateLiveRatesFlag|Description|RateDefinitionName

MERGE|RateDefinitionV2|Basic_Salary|Standard Rate Annualized|Annually|Annually|2020/01/01|US LDG|
Basic_Salary|ET|RateElement_001|VISION|USD|Regular Salary|A|Y|6|D|A|Y|Y|Y|Y|Basic_Salary|Basic_Salary

METADATA|RateContributorV2|EmploymentLevel|ContributorType|ContributorInputValueCode|EffectiveStartDate|
Feed|LegislativeDataGroupName|RateDefinitionCode|SequenceNumber|ReturnFullTimeRateFlag|SourceSystemId|
SourceSystemOwner
```

```
MERGE|RateContributorV2|Payroll Assignment|IV|Amount|2020/01/01|1|US LDG|Basic_Salary|1|Y|RateElement_001|
VISION
```

Related Topics

- Rates Based on Retained Grades
- Create Rate Definitions for Leave
- Generate HCM Rates

Loading Payroll Time Cards

Use HCM Data Loader to import time cards from a third-party time collection system.

You need to create elements, with the Timecard category, before you import time to Cloud payroll. The elements control how time information is processed in the payroll calculation. They also control how time is imported into payroll when a change is made to a timecard. Depending on your element configuration, timecard changes are imported into Payroll using either the 'Update' or 'Delete and Create' method.

Let's look at an example to understand these different methods. An employee submits a weekly timecard with 7.5 hours salary per day, Monday to Friday. The timecard is loaded into the payroll application. Subsequently, the employee updates Thursday's salary hours to 8.5 hours.

- **Update (Recommended):** When time is next loaded to payroll, only the 8.5 hours update is loaded.

  Note:
  - It is recommended you enable the 'update' solution for all time elements. This feature has several benefits, such as reducing the volume of time entries, and supporting costing updates at the time entry level.
  - When you enable the update feature you must load costing information using the time entry component.

- **Delete and Create:** When time is next loaded to payroll, the original 7.5 hours is deleted and a new time entry is created for 8.5 hours.

HDL Time Costing

You can load costing information to indicate how time should be distributed by the payroll costing process. The costing segment information can be loaded using the Time Entry or the Time Entry Property component.

Let's look at an example to understand these options. An employee enters 40 hours basic pay on their timecard and assigns cost account '123' to these hours. The timecard information is loaded into Payroll. The employee subsequently updates their timecard and changes the cost account from '123' to '444'.

You can load the corrected costing details using one of these options:

- **Costing at Time Entry (Recommended):** Load an update to change the cost account to '444' on the time entry records.
- **Costing at Time Entry Property:** Use HDL to delete the 40 hours on the time entry records and delete the time entry property records for cost account '123'. You must also create new time entry records for the 40 hours and new time entry property records to assign the hours to cost account '444'.

> **Note:** You must load costing information using the time entry component when you enable the time 'update' feature.

Costing on Time Entry:

It is recommended you load costing information for a timecard using the time entry component. This solution supports the following key features:

- You can create, update, and delete costing information for a timecard.
- Costing at time entry level can be used with the new time 'update' feature.
- You can rollback time files with costing information on the Time Entry component.

> **Note:** When you have a requirement to delete costing information such as when it's entered against an incorrect time entry, you must delete the time entry record including the costing information and create a new time entry record without the costing information.

Costing on Time Entry Property:

The Time Entry Property solution will continue to support costing information for backwards compatibility, but it is recommended you move to the Time Entry component solution.

> **Note:**

- You can only load costing on the Time Entry Property component when you are using the original 'Delete and Create' time solution.
- Only the 'Create' operation is supported when costing records are loaded using the Time Entry Property component. This also prevents you from rolling back a time batch that has costing information on Time Entry Property component. All operations (create, update, and delete) are supported for other types of time property details such as State, County, and City.

Related Topics

- Example of Loading Payroll Time Cards
- Process Time Entries in Payroll
- Create Time Card Elements for Time Entries

Example of Loading Payroll Time Cards

Use HCM Data Loader to import time cards from a third-party time collection system to the Payroll cloud.

Let's look at this example. Recently,Vision Corp has acquired a company and imports the acquired employees' approved time to Payroll for payments.

Note:

- When creating Payroll time cards, you have the option to pass rate information for a time entry. You can either pass the Rate Name, Rate Definition ID or Rate Value. If you pass the Rate Value, you must also include the Periodicity.
- The Payroll TimeCard object doesn't support date-effective updates.

This PayrollTimeCard.dat file uses source keys to import approved time entries into the Oracle Payroll Cloud:

```
METADATA|PayrollTimeCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
AssignmentId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|TimeCardId
MERGE|PayrollTimeCard|VISION|9999890|Vision Corporation US LDG|HDL034233|2020/01/11|2020/01/18|6
METADATA|TimeEntry|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|AssignmentId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|TimeCardInstanceId(SourceSystemId)|TimeType|Time|UnitOfMeasure|
TimeCardId|TimeEntryId|Periodicity|Factor|RateDefinitionId|RateValue|Segment1
MERGE|TimeEntry|VISION|8888812|Vision Corporation US LDG|HDL034233|2020/01/11|2020/01/11|9999890|
NB_ZHRX_OTL_CDRM|8.0|H_DECIMAL3|6|6|HOURLY|1.0|300100122876642|7.0|101
MERGE|TimeEntry|VIRX_OSION|8888813|Vision Corporation US LDG|HDL034233|2020/01/12|2020/01/12|9999890|
NB_ZHRX_OTL_CDRM|8.0|H_DECIMAL3|6|7|HOURLY|1.0|300100122876642|7.0|102
MERGE|TimeEntry|VISION|8888814|Vision Corporation US LDG|HDL034233|2020/01/13|2020/01/13|9999890|
NB_ZHRX_OTL_CDRM|8.0|H_DECIMAL3|6|8|HOURLY|1.0|300100122876642|7.0|101
MERGE|TimeEntry|VISION|8888815|Vision Corporation US LDG|HDL034233|2020/01/14|2020/01/14|9999890|
NB_ZHTL_CDRM|8.0|H_DECIMAL3|6|9|HOURLY|1.0|300100122876642|7.0|101
METADATA|TimeEntryProperty|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
AssignmentId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|PayrollTimeItemId(SourceSystemId)|TimeType|
PropertyName|PropertyValue|TimeCardId|TimeEntryId|TimeCardInstanceId(SourceSystemId)
MERGE|TimeEntryProperty|VISION|777712|Vision Corporation US LDG|HDL034233|2020/01/11|2020/01/11|8888812|
NB_ZHRX_OTL_CDRM|NB_ZHRX_OTL_CDRM_State|12|6|6|9999890
```

This example uses user keys to reference the time card data.

```
METADATA|PayrollTimeCard|LegislativeDataGroupName|AssignmentNumber|EffectiveStartDate|EffectiveEndDate|
TimeCardId
MERGE|PayrollTimeCard|Vision Corporation US LDG|E15CDRM|2020/01/11|2020/01/18|6
METADATA|TimeEntry|LegislativeDataGroupName|AssignmentNumber|EffectiveStartDate|EffectiveEndDate|TimeType|
Time|UnitOfMeasure|TimeCardId|TimeEntryId|Periodicity|Factor|RateName|RateValue|Segment1
MERGE|TimeEntry|Vision Corporation US LDG|E15CDRM|2020/01/11|2020/01/11|NB_ZHRX_OTL_CDRM|8.0|H_DECIMAL3|6|6|
HOURLY|1.0|COST_PROCESS_ELEMENT|7.0|101
METADATA|TimeEntryProperty|LegislativeDataGroupName|AssignmentNumber|EffectiveStartDate|EffectiveEndDate|
PropertyName|PropertyValue|TimeType|TimeCardId|TimeEntryId
MERGE|TimeEntryProperty|Vision Corporation US LDG|E15CDRM|2020/01/11|2020/01/11|NB_ZHRX_OTL_CDRM_State|12|
NB_ZHRX_OTL_CDRM|6|6
```

Related Topics

- Process Time Entries in Payroll

How You Roll Back Payroll Time Cards

You can roll back time card entries by using payroll flows and the Import and Load Data task.

Note: Only newly created payroll time cards that haven't been updated since they were created are rolled back.

Use Payroll Flow To Roll Back Time Cards

If you had used payroll flows to load time card, then you can roll it back using the flow's rollback feature. Use the **Initiate Data Loader** flow pattern to roll back time card entries that you uploaded using the flow pattern.

To initiate a payroll flow to upload time cards, complete these steps:

1. Create the PayrollTimeCard.dat file to load your time card entries. You must compress it and provide a file name of your choice.
2. Open **Submit a Flow**.
3. From the Legislative Data Group menu, select the required LDG.
4. Search for and click **Initiate Data Loader**.
5. Enter Payroll Flow, and in the Flow Parameter section, upload the compressed file.
6. Submit the flow and refresh until the flow status shows Completed.

To roll back the time entries, do these steps:

1. Open **View Flows** and search with the payroll flow name submitted earlier.
2. Click **Flow Name** and from the **Action** menu, select the rollback option.
3. Refresh until the flow status is changed to Corrected processes.

The application rolls back the newly created time cards.

Use Import and Load Data Task To Roll Back Time Cards

1. Open the **Data Exchange** work area.
2. Click **Import and Load Data**.
3. Search for and select the data set with the time card entries to roll back.
4. On the **Business Objects** table toolbar, click **Submit**
5. In the **Schedule Request** dialog box, **Action** field, select **Roll Back**.
6. Click **Submit**.

Example for Loading Absence Records

Absence records that are imported from Oracle Cloud Absences or third-party applications are stored and processed within the Oracle Cloud Payroll application. You can view the imported records by navigating to the Absence Entries region of the Calculation Entries page.

Using HCM Data Loader, you can create, update, and delete absence records. You can update and delete the absences that have:

- Not yet been processed by the payroll application
- Been processed by the payroll application
- Been partly processed by the payroll application

Create an Absence

Let's consider this example. An employee is sick from 01 Mar to 03 Mar. Their line manager applies for sickness absence in the company's third-party absence application. The absence administrator extracts the absence and sends it to Cloud Payroll for payment.

Update an Absence

Let's consider this example. An employee is sick from 01 Mar to 03 Mar. Their line manager applies for sickness absence in the company's third-party absence application. The absence administrator extracts the absence and sends it to Cloud Payroll for payment. However, the employee extends the sickness absence by one day.

Using HCM Loader, the line manager updates the absence record to include the additional absence of one day - 04 Mar. The Absence Administrator extracts the corrected absence information and sends it to the payroll application.

Related Topics

- Example of Loading Payroll Time Cards

How You Roll Back Payroll Absence Records

You can roll back absence records by using payroll flows and the **Import and Load Data** task.

Note: Only newly created absence records, that haven't been updated since they were created, will be rolled back.

Use Payroll Flow To Roll Back Absence Records

If you used payroll flows to load absence record, then you can roll it back using the flow's rollback feature. Use the **Initiate Data Loader** flow pattern to roll back absence entries that you uploaded using the flow pattern. To initiate a payroll flow to upload absence records, complete these steps:

1. Create the PayrollAbsenceRecord.dat file to load your absence entries. You must compress it and provide a file name of your choice.
2. Open **Submit a Flow**.
3. From the **Legislative Data Group** menu, select the required LDG and search for Initiate Data Loader.
4. Enter Payroll Flow and in the Flow Parameter section, upload the compressed file.
5. Submit the flow and refresh until the flow status shows Completed.

To roll back the absence entries, complete these steps:

1. Open **View Flows** and search with the payroll flow name submitted earlier.
2. Click **Flow Name**, and from the **Action** menu, select the **Rollback** option.
3. Refresh until the flow status is changed to Corrected processes.

The application rolls back the newly created absence records.

Use Import and Load Data Task To Roll Back Absence Records

1. Open the **Data Exchange** work area.
2. Click **Import and Load Data**.
3. Search for and select the data set with the absence entries to roll back.
4. Click **Submit** on the **Business Objects** table toolbar.
5. In the **Schedule Request** dialog box, **Action** field, select **Roll Back**.
6. Click **Submit**.

Guidelines for Loading Calculation Cards

Calculation cards capture values required for payroll calculation of some earnings and deductions, such as absence payments and involuntary deductions.

Various types of calculation cards exist. For example, UK calculation cards include those for statutory deductions, automatic enrollment in pensions, court orders and student loans, and benefits and pensions. Most legislations support the Calculation Card object. However, the types of calculation cards that are supported vary by legislation. This topic describes some general considerations that apply to all calculation cards when you load them using HCM Data Loader. For legislation-specific information, see *DOC ID 1504483.1*

Each calculation card has one or more components, and each component may have one or more sets of component details, or value definition overrides. For example, in the UK court orders and student loans calculation card, each component corresponds to a different type of court order. You enter information about that court order in the component details and value definition overrides.

Calculation cards are used mainly at the level of the payroll relationship. Depending on the legislation, they can also be used at the level of the payroll statutory unit or tax reporting unit.

Key Support

Calculation Card objects are integration-enabled. They support all key types that HCM Data Loader supports.

Identify Autogenerated Calculation Cards

Some calculation cards are autogenerated, and you're unlikely to know the source key values that uniquely identify the calculation card records.

You can supply user key values to identify these records, but this relies on you knowing the assignment number of your employee's assignment. If you don't know this, you can:

- Extract the autogenerated source key values using the Integration Object User Key Map Extract
- Optionally update the autogenerated source key

Create New Calculation Cards

If you're creating a new calculation card, you're recommended to use source keys to uniquely identify the records being loaded. This simplifies the file shape and allows you to easily identify your calculation card records when you need to maintain them.

Update Existing Calculation Cards

If you know your source key values, it is recommended that you use these to reference the records to be updated; user keys are also supported.

When updating existing values for components, the effective date of the component and calculation value definitions in your file or spreadsheet must be the original start date of the existing card component. The Enterable Calculation Values include the actual date of the value change.

If required, the Integration Object User Key Map Extract can be used to extract all key values for existing calculation card records.

You can find details of the parent source key attributes and alternate user key attributes using the View Business Objects task.

Change Logical Start and End Dates

In general, you can change the first effective start date and last effective end date of an existing calculation card. Include the **ReplaceFirstEffectiveStartDate** and **ReplaceLastEffectiveEndDate** attributes in your METADATA line, as appropriate, and set them to **Y**. Provide the new values on the **EffectiveStartDate** and **EffectiveEndDate** attributes.

Note: Legislation-specific validations may mean that some changes to logical start and end dates aren't valid.

Delete Calculation Cards

You can delete all components of the Calculation Card object using HCM Data Loader. When you delete a Calculation Card component, its child components are deleted automatically. You can also delete individual child components.

Related Topics

Example of Loading Wage Basis Rules

Use HCM Data Loader to load wage basis rules to determine the earnings that are subjected to a deduction.

To illustrate how wage basis rules affect a tax calculation, let's look at an example where an employee's earnings included in the wage basis vary, depending upon where the employee lives.

Brittany is a salesperson who receives a salary of 2,000 each month. Brittany also has a company car she drives. She is responsible for reporting her personal use of the company car in order to be taxed correctly.

Brittany works in her company's East Coast district. She spends 50 percent of her time in New York and 50 percent of her time in Pennsylvania. The State of New York taxes Brittany for her personal use of the company car; however, Pennsylvania doesn't.

Last month, Brittany reported personal use that equated to 100 (50 personal use in New York and 50 personal use in Pennsylvania).

In New York, the imputed earnings, such as personal use of company car amounts are included in the taxable wages. Whereas, in Pennsylvania, these earnings aren't included in taxable wages.

This table shows the tax calculations that apply for each region.

Region	Earnings in Salary	Eligible Imputed Earnings	Taxable Income	Deduction Amount
New York	1000	50	1050	35
Pennsylvania	1000	50 - Exempt	1000	30

Here are the wage basis rule for this tax calculation.

Region (Reference Value)	Primary Classification	Secondary Classification	Use in Wage Basis?
New York	Standard Earnings	Regular	Y
New York	Imputed Earnings	Personal Use of Company Car	Y
Pennsylvania	Standard Earnings	Regular	Y
Pennsylvania	Imputed Earnings	Personal Use of Company Car	N

Use this TaxabilityRule.dat to load the wage basis rules.

```
METADATA|TaxabilityRule|EffectiveStartDate|LegislativeDataGroupName|PayrollComponent|PrimaryClassification|
SecondaryClassification|Context1|UsageType
MERGE|TaxabilityRule|2011/1/10|Vision Corp|State Tax|Standard Earnings|Regular|NY|P
MERGE|TaxabilityRule|2011/1/10|Vision Corp|State Tax|Imputed Earnings|Personal Use of Company Car|NY|P
MERGE|TaxabilityRule|2011/1/10|Vision Corp|State Tax|Standard Earnings|Regular|PA|P
MERGE|TaxabilityRule|2011/1/10|Vision Corp|State Tax|Imputed Earnings|Personal Use of Company Car|PA|N
```

Related Topics

- Wage Basis Rules

Element Entries

Overview of Loading Element Entries

You use element entries to capture earnings, deductions, absences, and basic benefit details for an employee assignment. For example, you can create element entries for an employee's overtime hours or medical premium deduction amount.

The element entry values hold the necessary values for the element type. For example, a salary entry can contain the salary value and the salary frequency.

HCM Data Loader provides these two business objects for loading element entries:

Object	What It Loads
Element Entry	Element entries and element entry values. **Note:** To load costing data for element entries created with the Element Entry object, use the Cost Allocation and Cost Allocation Account objects.

Object	What It Loads
Element Entry Costing	Element entries and element entry values along with the costing information at element entry level.

Element Entry With Costing

Here are some guidelines that you must consider when loading element entries using the Element Entry with Costing object.

- Unlike most HCM Data Loader objects, the Element Entry with Costing object isn't hierarchical. The entry values and costing segments are supplied on the same line as the element entry information.

  The costing information is optional. Using the Element Entry with Costing object, you can create just element entries too.
- You can't update, end-date or delete element entries using the Element Entry with Costing object.
- You can't rollback the new element entries with costing information.
- You can't load date effective history of an element.
- You can't load different effective start dates for element entry and costing. The application uses the same effective date as the Element Entry start date and Costing start date.
- The entry sequence of the element entry is automatically generated for you. The Create Entry Sequence is unique within the dat file for an entry. The application reads the existing element entry sequence and automatically increments the sequence number using the number provided in the dat file. You don't have to review the existing entries and provide the next sequence number in the dat file that you do in individual element entry object.
- If you reload the same file with only user keys, the application will create a new element entry on every iteration. This behavior is because `CreateEntrySequence` isn't stored in the application and doesn't uniquely identify the record.
- Supply a source key when creating element entries, as the element entry sequence number is generated for you. Also, you wouldn't know the user key in order to maintain the element entry or costing records.
- You must use the Cost Allocation and Cost Allocation Account objects to maintain element entry costing that you created using the Element Entry with Costing objects. The source key supplied when creating the element entry can be used to identify the costing record to maintain.
- You must use the Element Entry object to update element entries created using the Element Entry with Costing object. The source key supplied when creating the element entry can be used to identify the element entry. To identify the source keys for entry values, use this combination - **`<EntrySourceId><InputValueBaseName>`**.

Note: You must replace the spaces with underscore in `InputValueBaseName`.

Related Topics

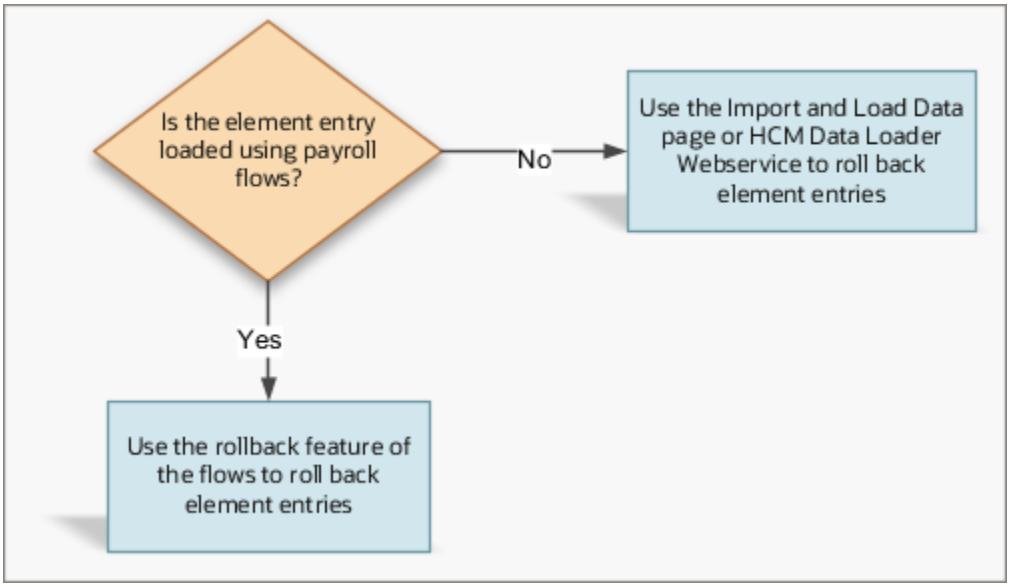
- Enable Automatic, Multiple, or Additional Element Entries Options
- Options to Determine an Element's Latest Entry Date
- Default Values for Element Entries

Overview of Rolling Back Element Entries

You can only roll back new element entries that you loaded using the Element Entry object. To roll back element entries, there's an additional constraint that there should be no changes made to the object since it was created.

If HCM Data Loader identifies that the element entry has been updated, the rollback fails. The application doesn't make an attempt to roll back updates to an existing element entry. Also, it doesn't roll back the deletion of an existing element entry.

This figure illustrates how you can roll back element entries depending upon whether the entries were loaded using payroll flows or not.



If you submit an element entry using the Initiate Data Loader flow, then you can roll it back using one of the features:

- Payroll Flows
- The Import and Load Data page
- The HCM Data Loader web service

Alternatively, if you didn't submit your element entry using a flow pattern, then use the Import and Load Data page to roll them back.

> **Note:** Let's assume you loaded Element Entry, Cost Allocation, and Cost Allocation account using a single data set from the Import and Load data page. To roll back the objects in that data set, you must roll back the objects in the reverse order. In other words, you must roll back Cost Allocation Account first, then Cost Allocation, and finally Element Entry.

Rollback and Events

When an element entry is rolled back, the application also rolls back any unprocessed events and notifications.

Consider these scenarios that explain the impact of a roll back on events.

- If the event has an unprocessed status, then the application removes the event. If the event is already processed, then the application lets it stay.
- If you created an element entry, then application removes the event. The application doesn't remove other types of element entry events, such as an update element entry event.
- If you updated an element entry, then the application doesn't delete that event.

Related Topics

- Enable Automatic, Multiple, or Additional Element Entries Options
- Options to Determine an Element's Latest Entry Date
- Default Values for Element Entries

How You Roll Back Element Entries

You can roll back element entries by using payroll flows and the **Import and Load Data** task.

Use Payroll Flow To Roll Back Entries

If you used payroll flows to load an element entry, then you can roll it back using the flow's rollback feature. Use the **Initiate Data Loader** flow pattern to roll back element entries that you uploaded using the flow pattern.

These steps describe how you initiate a payroll flow to upload element entries.

1. As the first step, you create the `ElementEntry.dat` file to load your element entries and its values. Also, you must compress it to a file name of your choice.
2. Using the **File Import and Export** task, you upload the dat file to the Oracle Web Center Content Server. Enter the account as `hcm/dataloader/import`. You must make a note of the Content ID.
3. From the Checklist or Data Exchange work area, you submit the **Initiate Data Loader** flow to load and submit the element entries data set. When submitting the flow, provide the Content ID from the previous step.

To roll back your element entry data, on the Payroll Flow page, select the **Initiate Data Loader** flow. Select **Mark as Incomplete** from the **Actions** menu. In the Warning dialog box, click **Continue**. The application rolls back the newly created element entries.

Use Import and Load Data Task To Roll Back Entries

1. Navigate to the Data Exchange work area.
2. Select **Import and Load Data**.
3. Search for and select the data set with the element entries to roll back.
4. Click **Total Objects**.
5. On the Object Status page, click **Schedule Request**.
6. In the Schedule Request dialog box, **Action** field, select **Roll Back**.
7. Click **Submit**.

Related Topics

- Enable Automatic, Multiple, or Additional Element Entries Options
- Default Values for Element Entries

Example of Loading a Recurring Element Entry With an Open End Date

Lisa Jones has been asked to work from the company's headquarters from 01-Apr-2019 until further communication. As part of this assignment, she's paid a Commutation Allowance of 1000 per pay period.

Load the element entry details for Lisa, as shown in this table:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Commutation Allowance
Assignment Number	E3141464
Start Date	01-Apr-2019

Load the element entry values, as shown in these tables:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Commutation Allowance
Assignment Number	E3141464
Start Date	01-Apr-2019
Input Value Name	PRD
Screen Entry Value	1000

This ElementEntry.dat file creates the element entry values.

```
METADATA|ElementEntry|SourceSystemOwner|SourceSystemId|EffectiveStartDate|EffectiveEndDate|
LegislativeDataGroupName|ElementName|EntryType|CreatorType|HrAssignmentId(SourceSystemId)
```

```
MERGE|ElementEntry|VISION|12351_COM_ALW|2019/06/01|4712/12/31|Vision Corp|Commutation Allowance|E|H|
12351_ASG
METADATA|ElementEntryValue|SourceSystemOwner|SourceSystemId|EffectiveStartDate|EffectiveEndDate|
LegislativeDataGroupName|ElementName|InputValueName|ScreenEntryValue|ElementEntryId(SourceSystemId)
MERGE|ElementEntryValue|VISION|12351_COM_ALW_PRD|2019/06/01|4712/12/31|Vision Corp|Commutation Allowance|
PRD|1000|12351_COM_ALW
```

Note:

- You don't need to supply the MultipleEntryCount attribute as source keys to uniquely identify the records.
- When loading a recurring element entry with an end date, supply the EffectiveEndDate attribute on all file lines for the element entry.

Related Topics

- Enable Automatic, Multiple, or Additional Element Entries Options
- Options to Determine an Element's Latest Entry Date

Example of How To End Date a Recurring Element Entry

To update the last effective end date, include the `ReplaceLastEffectiveEndDate` attribute in your METDATA line. Provide a Y value to indicate that the `EffectiveEndDate` supplied is a change to the existing logical end date of your record.

You must supply a record for the element entry and every element entry value.

Let's consider the same example where Lisa Jones was asked to work from the company's headquarters from 01-Apr-2019 until further communication. When Lisa's deputation ended on 30-Jun-2019, the company decides to stop the allowance that it gave as part of this assignment.

Use this ElementEntry.dat file to end date the Commutation Allowance effective from 30-Jun-2019 and the element entry values. For more information on how to create a commutation allowance entry, refer to the topic: Loading a Recurring Element Entry with an Open End Date.

```
METADATA|ElementEntry|SourceSystemOwner|SourceSystemId|EffectiveStartDate|EffectiveEndDate|
ReplaceLastEffectiveEndDate
MERGE|ElementEntry|VISION|12351_COM_ALW|2019/06/01|2019/06/30|Y
```

Related Topics

- Enable Automatic, Multiple, or Additional Element Entries Options
- Options to Determine an Element's Latest Entry Date
- Default Values for Element Entries

Example of Loading Non-Recurring Element Entries

You can use HCM Data Loader to load non-recurring element entries. The application creates the non-recurring element entries for a specific period, usually the payroll period.

When you create element entry for non-recurring elements, you must increment the value of `MultipleEntryCount` for each entry of the same assignment and element.

Example: Lynda Jones with assignment number E3141464 already has an element entry for Transportation Allowance of $30 in the semi-monthly payroll period 01-Oct-2020 to 15-Oct-2020. The `MultipleEntryCount` for this entry is 1.

If you load the Transportation Allowance for Lynda, you need to increment the `MultipleEntryCount` for the next records.

Let's consider this scenario. Lynda Jones has traveled from corporate office to project site on 10-Oct-2020 and 12-Oct-2020. She must be paid a transportation allowance of $40 and $50 respectively.

Note that the two dates 10-Oct-2020 and 12-Oct-2020 fall within the same payroll period from 01-Oct-2020 to 15-Oct-2020.

Load the element entry details for 10-Oct-2020, as shown in this table:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	10-Oct-2020
MultipleEntryCount	2

Load the element entry values for 10-Oct-2020, as shown in these tables:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	10-Oct-2020
Input Value Name	Amount
Screen Entry Value	40

Load the element entry details for 12-Oct-2020, as shown in this table:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	10-Oct-2020
MultipleEntryCount	3

Load the element entry values for 12-Oct-2020, as shown in these tables:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	12-Oct-2020
Input Value Name	Amount
Screen Entry Value	50

The ElementEntry.dat file creates the element entry values.

```
METADATA|ElementEntry|CreatorType|EffectiveStartDate|ElementName|LegislativeDataGroupName|EntryType|
AssignmentNumber|MultipleEntryCount
MERGE|ElementEntry|F|2020/10/10|Transportation Allowance|Vision Corp|E|E3141464|2
MERGE|ElementEntry|F|2020/10/12|Transportation Allowance|Vision Corp|E|E3141464|3
METADATA|ElementEntryValue|EffectiveStartDate|ElementName|LegislativeDataGroupName|InputValueName|
ScreenEntryValue|AssignmentNumber|MultipleEntryCount|EntryType
MERGE|ElementEntryValue|2020/10/10|Transportation Allowance|Vision Corp|Overridden Pay Value|40|E3141464|2|E
MERGE|ElementEntryValue|2020/10/12|Transportation Allowance|Vision Corp|Overridden Pay Value|50|E3141464|3|E
```

Example of Deleting an Element Entry

Use HCM Data Loader to delete element entries. As you can't recover deleted records, you must take caution when deleting them. Don't attempt to delete an element entry and create an entry for the same element, assignment and date in the same file.

John Gorman is a worker in the Administration Department of Vision Corp. Assuming that he belongs to the Sales department, the company wrongly awards an incentive to him. Using HCM Data Loader, you can delete the sales incentive element entry.

This file deletes these element entry values.

```
METADATA|ElementEntry|CreatorType|EffectiveEndDate|EffectiveStartDate|ElementName|LegislativeDataGroupName|
EntryType|AssignmentNumber|MultipleEntryCount
DELETE|ElementEntry|F|4712/12/31|2019/04/01|Incentive Compensation|Vision Corp|E|E3141464|1
```

Delete Element Entries Created With Element Entry With Costing Object

Use the Element Entry object to delete element entries that were created with the `ElementEntrywithCosting` object.

Let's consider this example:

Lisa Jones is posted to a project location where there is no company provided accommodation available. Hence, the company provided her a housing allowance of 2000 per pay period. This element entry is costed to the cost center Company Head Quarters for which the Cost Center value is 100 and Funding Source is 200. Using the `ElementEntryWithCosting` object, you create the element entry values with costing information. For more information on how to create the element entry values using the `ElementEntryWithCosting` object, refer to this topic.

- Example of Loading a Recurring Element Entry With an Open End Date along with Costing Information

This dat file deletes the element entry created with costing information:

```
METADATA|ElementEntry|SourceSystemOwner|SourceSystemId|EffectiveStartDate
DELETE|ElementEntry|VISION|E3141464_HSEALW_190401|2019/04/01
```

Related Topics

- Example of Loading a Recurring Element Entry With an Open End Date along with Costing Information
- Enable Automatic, Multiple, or Additional Element Entries Options
- Options to Determine an Element's Latest Entry Date
- Default Values for Element Entries

Example of Loading Multiple Occurrences of the Same Element Entry in the Same Time Period

Bob traveled from his work location to client's office on 04-Jun-2019 and 11-Jun-2019 for meetings with clients. He is reimbursed 40 and 50 respectively as Transport Fare along with his salary on 15-Jun-2019.

The first reimbursement of 40 should go to the cost center 100 and the second reimbursement of 50 should go to the cost center 200. He is a as part of the semi-monthly payroll processing cycle.

Using HCM Data Loader, load these element entry values. This is the first occurrence of the element entry and the CreateEntrySequence attribute will have value 1.

This table shows the first element entry value along with costing details:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	04-Jun-2019
End Date	04-Jun-2019
Cost Center	100

This table shows the element entry values to load:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	04-Jun-2019
End Date	04-Jun-2019
Input Value Name	Amount
Screen Entry Value	40

This table shows the second element entry value along with costing details:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	11-Jun-2019

Attribute	Value
End Date	11-Jun-2019
Cost Center	200

This table shows the element entry values to load:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	11-Jun-2019
End Date	11-Jun-2019
Input Value Name	Amount
Screen Entry Value	50

This ElementEntryWithCosting.dat file creates multiple element entries, entry values along with the costing information.

```
METADATA|ElementEntryWithCosting|EffectiveEndDate|EffectiveStartDate|ElementEntryId|ElementTypeId|
ElementName|AssignmentId(SourceSystemId)|AssignmentNumber|Reason|Subpriority|EntryType|CreateEntrySequence|
SourceSystemOwner|SourceSystemId|InputValueName1|InputValueName2|InputValueName3|InputValueName4|
InputValueName5|ScreenEntryValue1|ScreenEntryValue2|ScreenEntryValue3|ScreenEntryValue4|ScreenEntryValue5|
Segment1|Segment2|Segment3|Segment4|Segment5|Segment6|Segment7|Segment8
MERGE|ElementEntryWithCosting|2019/06/04|2019/06/04|||Transportation Allowance||E3141464|||E|1|||Amount|||||
40|||||||100||||
MERGE|ElementEntryWithCosting|2019/06/11|2019/06/11|||Transportation Allowance||E3141464|||E|2|||Amount|||||
50|||||||200||||
```

Related Topics

- Enable Automatic, Multiple, or Additional Element Entries Options
- Options to Determine an Element's Latest Entry Date
- Default Values for Element Entries

Element Entry With Costing

Overview of Loading Element Entry with Costing

Use the **Element Entry with Costing** object to load the element entries along with the costing information at the element entry level.

The load validates whether the costing segment value is active or not as on the load date. The load fails, if the costing segment value isn't valid as on the load date. Unlike many HCM Data Loader objects, the Element Entry with Costing object isn't hierarchical. The entry values and costing segments are supplied on the same line as the element entry information. The costing information is optional. Using the Element Entry with Costing object, you can create just element entries too.

Consider these guidelines when **creating** new element entries using the Element Entry with Costing object.

- You must supply the CreateEntrySequence attribute with a value that's unique within the ElementEntryWithCosting.dat file for the Element Type and Assignment the entry is for. The entry sequence on the created element entry is then automatically generated for you by incrementing the largest existing element entry sequence for the combination of element type and assignment number.
- It's always recommended that you supply source keys when creating new records. The source key supplied can then be used to identify the element entry, entry values and costing records when maintaining them.
- If you don't supply source keys and reload the same file, new records will be created.
- You can't create date-effective history using this object.
- You can't specify different effective start dates for the element entry and costing using the object.
- You can't end-date element entries using the Element Entry with Costing object.
- You can’t do a replace end date operation using the Element Entry with Costing object.
- Supply #NULL as entry value for an input value if you’re using HCM Spreadsheet data loader and wanted to use the default value from the element eligibility input value level. Otherwise, default values from element input value level will be used if you leave it blank. You can also supply the desired value if you want a specific value other than the default from element input value or element eligibility input value.

  For example, element input value has a default of 1000 and element eligibility input has the default of 2000 in the element setup. When HCM Spreadsheet template is generated for this element, the row will be generated with 1000 by default. You need to supply #NULL to this entry value in order to use 2000 from element eligibility input value.

Consider these guidelines when **updating** existing element entries that were created using the Element Entry with Costing object.

- To correct or update the element entry and costing information created with this object, use the Element Entry with Costing object . The effective date supplied in the dat file is used for updating or correcting both element entry and costing information. If you have a requirement to use separate effective dates for element entry and costing, you should use the respective individual objects to perform the operation.
- To maintain costing records created with the Element Entry with Costing object, use the CostAllocationV3 and CostAllocationAccountV3 objects . The source key supplied when creating the Element Entry is the same used to identify the costing records.

- To correct or update element entries that you created with the Element Entry with Costing object, and need a separate effective date for entry and costing information, use the ElementEntry object . The source key supplied when creating the Element Entry is the same used to identify the Element Entry records. Element Entry Value records use with same source system ID but with the Input Value Base Name appended. Replace any spaces in the Input Value Base Name with underscores.

  Note: You must replace the spaces with underscores in InputValueBaseName.

Consider this when **deleting** element entries using the Element Entry with Costing object.

- To uniquely identify the record to delete, supply either the source key or the EntrySequence along with the other user key attributes.

Example of Loading a Recurring Element Entry With an Open End Date along with Costing Information

Lisa Jones is posted to a project location where there is no company provided accommodation available. Hence, the company provided her a housing allowance of 2000 per pay period.

This element entry is costed to the cost center Company Head Quarters for which the Cost Center value is 100 and Funding Source is 200.

Load the element entry details for Lisa, as shown in this table:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Housing Allowance
Assignment Number	E3141464
Start Date	01-Apr-2019
Cost Center	100
Funding Source	200

Load the element entry values, as shown in these tables:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Housing Allowance

Attribute	Value
Assignment Number	E3141464
Start Date	01-Apr-2019
Input Value Name	Amount
Screen Entry Value	1000

This ElementEntryWithCosting.dat file creates the element entry values with costing information.

```
METADATA|ElementEntryWithCosting|SourceSystemOwner|SourceSystemId|ElementName|EffectiveStartDate|
EffectiveEndDate|AssignmentId(SourceSystemId)|EntryType|CreateEntrySequence|InputValueName1|
ScreenEntryValue1|InputValueName2|ScreenEntryValue2|Segment1|Segment2|Segment3|Segment4|Segment5|Segment6|
Segment7|Segment8
MERGE|ElementEntryWithCosting|VISION|E3141464_HSEALW_190401|Housing Allowance|2019/04/01||E3141464|E|1|
Amount|1000||200|100|
```

Note: Let's assume that you're loading an `ElementEntryWithCosting.dat` file to create 50 element entries. The HCM Data Loader application loads the valid element entries. However, some of the 50 element entries fail. You correct the underlying issue and now need to submit the load process again to load the previously failed records. On the Import and Load Data page, select your data set and select the Element Entry With Costing business object within it. Click Submit to submit the process with an action of Load. If you're using HCM Spreadsheet Data Loader, click **Upload** to submit previously failed records. If you import the file again after supplying the user keys, you'll get new element entries, provided the element type supports multiple entries in a period. You're recommended to supply source keys when using HCM Data Loader. The source key can then be used to identify the element entry, entry value, or costing record when you want to maintain or delete it. For the load to be successful, the costing segment value must be active as on the load date.

Related Topics

- Overview of Loading Element Entries
- Overview of Rolling Back Element Entries
- Example of Loading Multiple Occurrences of the Same Element Entry in the Same Time Period

Example of Loading Multiple Occurrences of the Same Element Entry in the Same Time Period

Bob traveled from his work location to client's office on 04-Jun-2019 and 11-Jun-2019 for meetings with clients. He is reimbursed 40 and 50 respectively as Transport Fare along with his salary on 15-Jun-2019.

The first reimbursement of 40 should go to the cost center 100 and the second reimbursement of 50 should go to the cost center 200. He is a as part of the semi-monthly payroll processing cycle.

Using HCM Data Loader, load these element entry values. This is the first occurrence of the element entry and the CreateEntrySequence attribute will have value 1.

This table shows the first element entry value along with costing details:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	04-Jun-2019
End Date	04-Jun-2019
Cost Center	100

This table shows the element entry values to load:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	04-Jun-2019
End Date	04-Jun-2019
Input Value Name	Amount
Screen Entry Value	40

This table shows the second element entry value along with costing details:

Attribute	Value
Legislative Data Group	Vision Corp

Attribute	Value
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	11-Jun-2019
End Date	11-Jun-2019
Cost Center	200

This table shows the element entry values to load:

Attribute	Value
Legislative Data Group	Vision Corp
Element Name	Transportation Allowance
Assignment Number	E3141464
Start Date	11-Jun-2019
End Date	11-Jun-2019
Input Value Name	Amount
Screen Entry Value	50

This ElementEntryWithCosting.dat file creates multiple element entries, entry values along with the costing information.

```
METADATA|ElementEntryWithCosting|EffectiveEndDate|EffectiveStartDate|ElementEntryId|ElementTypeId|
ElementName|AssignmentId(SourceSystemId)|AssignmentNumber|Reason|Subpriority|EntryType|CreateEntrySequence|
SourceSystemOwner|SourceSystemId|InputValueName1|InputValueName2|InputValueName3|InputValueName4|
InputValueName5|ScreenEntryValue1|ScreenEntryValue2|ScreenEntryValue3|ScreenEntryValue4|ScreenEntryValue5|
Segment1|Segment2|Segment3|Segment4|Segment5|Segment6|Segment7|Segment8
MERGE|ElementEntryWithCosting|2019/06/04|2019/06/04|||Transportation Allowance||E3141464|||E|1|||Amount|||||
40|||||||100||||
MERGE|ElementEntryWithCosting|2019/06/11|2019/06/11|||Transportation Allowance||E3141464|||E|2|||Amount|||||
50|||||||200||||
```

Related Topics

- Enable Automatic, Multiple, or Additional Element Entries Options
- Options to Determine an Element's Latest Entry Date
- Default Values for Element Entries

Maintain Element Entries Created with Element Entry With Costing

To update or correct the element entries along with the costing information when the effective date is same for both element entry and costing, you can use the `ElementEntrywithCosting` object.

If you need to use separate effective dates for element entry and costing while updating, use the respective individual objects, such as ElementEntry for updating or correcting the Element Entry information and CostAllocationV3 and CostAllocationAccountV3 for Costing Information.

Let's consider, after loading a recurring element entry with an open end date along with costing information, you realize that both the Funding Source and Amount were incorrect on the original element entry.

To correct Amount and Funding Source using the ElementEntrywithCosting object, use the ElementEntrywithCosting.dat file:

```
METADATA|ElementEntryWithCosting|EffectiveEndDate|EffectiveStartDate|ElementName|EntryType|AssignmentNumber|
EntrySequence|InputValueName1|ScreenEntryValue1|Segment1|Segment2
MERGE|ElementEntryWithCosting||2019/04/01|Housing Allowance|E|E3141464|1|Amount|2500|100|300
```

Let's say, you want to increase the housing allowance for Lisa Jones effective 01-April-2020. You would like to charge the amount to Cost Center 200 and Funding Source 400. In this scenario, you can update the element entry and costing information using the ElementEntrywithCosting object with the respective date.

To update the Amount, Cost Center and Funding Source using ElementEntrywithCosting object, use the ElementEntrywithCosting.dat file:

```
METADATA|ElementEntryWithCosting|EffectiveEndDate|EffectiveStartDate|ElementName|EntryType|AssignmentNumber|
EntrySequence|InputValueName1|ScreenEntryValue1|Segment1|Segment2
MERGE|ElementEntryWithCosting||2020/04/01|Housing Allowance|E|E3141464|1|Amount|3500|200|400
```

To correct amount entry value supply the ElementEntry.dat file:

```
METADATA|ElementEntry|SourceSystemOwner|SourceSystemId|EffectiveStartDate|EffectiveEndDate
MERGE|ElementEntry|VISION|E3141464_HSEALW_190401|2021/04/01|4712/12/31
METADATA|ElementEntryValue|SourceSystemOwner|SourceSystemId|EffectiveStartDate|EffectiveEndDate|
ScreenEntryValue|ElementEntryId(SourceSystemId)
MERGE|ElementEntryValue|VISION|E3141464_HSEALW_190401_Amount|2021/04/01|4712/12/31|1000|
E3141464_HSEALW_190401
```

Note: The source key from the ElementEntryWithCosting file is used to identify the element entry. To identify the element entry value, the same source key is used appended with the input value name.

To correct a costing segment, supply the CostAllocationAccountV3.dat file.

```
METADATA|CostAllocationAccountV3|SourceSystemOwner|SourceSystemId|EffectiveDate|ElementCode|
LegislativeDataGroupName|SourceSubType|AssignmentId(SourceSystemId)|ElementEntryId(SourceSystemId)|
SourceType|Segment4
```

```
MERGE|CostAllocationAccountV3|VISION|E3141464_HSEALW_190401|2021/04/01|EC_US_SEARMY_ELEMENT|ZHRX_USVS_ST LDG
 One|COST|E3141464|E3141464_HSEALW_190401|EE|300
```

Note: The source key from the ElementEntryWithCosting file is used to identify the cost allocation account record.

Example of Deleting an Element Entry with Costing

Use HCM Data Loader to delete element entries with costing. As you can't recover deleted records, you must take caution when deleting them.

Don't attempt to delete an element entry with costing and create it for the same element, assignment and date in the same file.

Lisa Jones is posted to a project location where there's no company provided accommodation available. Hence, the company provided her a housing allowance of 2000 per pay period. This element entry is costed to the Company Head Quarters cost center for which the Cost Center value is 300 and Funding Source is 400. Assuming that she worked in a different project location which doesn't have the housing allowance perquisite. Using HCM Data Loader, you can delete housing allowance entry along with the costing information.

This ElementEntryWithCosting.dat file uses the source key to delete the element entry and costing records created in the **Loading a Recurring Element Entry** example.

```
METADATA|ElementEntryWithCosting|SourceSystemOwner|SourceSystemId|EffectiveStartDateDELETE|
ElementEntryWithCosting|VISION|E3141464_HSEALW_190401|2021/04/0
```

If using user keys, you must use the EntrySequence attribute, not CreateEntrySequence, to delete the records. This ElementEntryWithCosting.dat file uses the user key to delete the element entry and costing records created in the **Loading a Recurring Element Entry** example.

```
METADATA|ElementEntryWithCosting|ElementName|EffectiveStartDate|EffectiveEndDate|AssignmentNumber|
EntrySequence|EntryTypeMERGE|ElementEntryWithCosting|Housing Allowance|2021/04/01||E3141464|1|E
```

How You Roll Back Element Entry with Costing information

You can only roll back new element entries with costing information that you loaded using the Element Entry with Costing object. To roll back, there's an additional constraint that there should be no changes made to the object since it was created.

When you submit roll back, all records that were created successfully are rolled back. You can't roll back individual records. If HCM Data Loader identifies that an element entry or costing information has been updated, the rollback fails for that record.

You can roll back element entries with costing information by using payroll flows or the Import and Load Data task.

Use Payroll Flow to Roll Back Entries

If you used payroll flows to load an element entry with costing file, then you can roll it back using the flow's rollback feature. Use the Initiate Data Loader flow pattern to roll back element entries with costing that you uploaded using the flow pattern.

These steps describe how you initiate a payroll flow to upload element entries.

1. Create the ElementEntryWithCosting.dat file to load your element entries with costing. Also, you must compress it to a file name of your choice.
2. Using the File Import and Export task, upload the dat file to the Oracle Web Center Content Server. Enter the account as hcm/dataloader/import. Don't forget to make a note of the Content ID.
3. From the Checklist or Data Exchange work area, you submit the Initiate Data Loader flow to load and submit the element entries data set. When submitting the flow, provide the Content ID from the previous step.

   To roll back your element entry with costing data, on the Payroll Flow page, select the Initiate Data Loader flow. Select Mark as Incomplete from the Actions menu. In the Warning dialog box, click Continue. The application rolls back the newly created element entries.

Use Import and Load Data Task to Roll Back Entries

1. Navigate to the Data Exchange area.
2. Select **Import and Load Data**.
3. Search for and select the data set with the element entries with costing to roll back.
4. In the Business Object table, select the Element Entry with Costing row.
5. Click **Submit** on the table toolbar.
6. On the Schedule Request dialog box page, Action field, select **Roll Back**.
7. Click **Submit**.

> **Note:**
> - You need to submit rollbacks with the same number of threads as the initial load, to avoid delays.
> - You can initiate rollbacks with higher number of threads from the Data Exchange UI at the Business Object level, in the reverse order sequentially – CostAccount, CostInfo and ElementEntry.

Generate Spreadsheet Templates for Element Entry with Costing

As a Payroll manager or an administrator, you can generate the HCM Spreadsheet Data Loader template for a single element or a group of elements using the Generate HCM Spreadsheet Data Loader Template payroll flow.

Using the Generate HCM Spreadsheet Data Loader Template payroll flow, you can automatically configure spreadsheet templates for bulk-loading element entries, based on your elements definition.

Here are some points to consider when generating spreadsheet templates:

- You can access generated templates through Spreadsheet Templates task in **Data Exchange** work area
- The name of the template is auto generated and is represented as <Element_Name> followed by the date and time stamp. For example: CRFL RRF EARNINGS US_20210806_110234 where CRFL RRF EARNINGS US is the element name and 20210806_110234 is the date and time stamp
- It's recommended that you rename the auto generated template name to help the users who access it
- You need to assign the roles that should have access to the template and the data set access that each of those roles have

- Generated templates are created in Draft mode. You need to set the Status to **Active** for the template to be available to the roles you've assigned to it
- The template can be generated for standard earnings, supplemental earnings, voluntary deductions and pretax deductions without any secondary classification. The category must be **Standard** for all these classifications.

There are three parameters in this flow:

Parameters	What you need to know
Business Object Name	You need to select the **Element Entry with Costing** value to generate the template for this object. This parameter currently has a single value.
Entity Type	This parameter has two values - Element Group and Element. You can generate the spreadsheet template for multiple elements using Element Group value in this parameter. Create element group using Object Group task with the list of elements for which the spreadsheet template to be generated. You can generate spreadsheet template for a single element using Element value in this parameter.
Entity Name	The list of values in this parameter is dependent on the Entity Type parameter. If you select Element Group, it shows the list of element groups that are created using Object Group task. If you select Element, it will show the list of elements.

Once you have selected the appropriate values and the submitted the flow, it submits two sub-flows. One flow validates the data selected and the second flow creates the spreadsheet template.

After the successful completion of both flows, for the successfully created elements, you can view the generated template names on the Process Results page. To see if any template wasn't created for an element, see Messages.

Note: Ensure that **Process** is selected in the **View By** option to view the messages for the submitted flow.

17 Initializing and Adjusting Balances

Overview of Balances

Overview of Balances

This topic describes what a payroll balance is and what are its primary components.

A payroll balance is defined as the accumulation of values over a specific period of time. The three main components of a balance are:

- Name
- Components that feed it (called balance feeds)
- Dimensions that define its unique characteristics

For example, the Regular balance is the accumulation of all Standard Earnings processed in a payroll run. Dimensions allow you to view the value of a balance based on a predefined combination of time (for example, Period-to-Date or Year-to-Date), employment level (assignment or payroll relationship), and context (required for specific balances only, such as tax reporting unit, element, or payroll). For example, you might want to view Gross Earnings for one employee for the current pay period, or as of year-to-date. Please note that these context values referred to in the balance initialization process and batch loader are also referred to as References in the Balance View User Interface (UI)).

Here are the various components of balances:

Balance Name

This is the actual name of the balance, for example Gross Earnings. Balance types always have a numeric unit of measure, and in some instances a currency code.

Balance Feed

Balance feeds define the element input values that make up a balance. Balance feeds can either add to (+) or subtract from (–) a balance.

Balance Dimension

Balance dimensions define the time period, employment relationship, and context from which to calculate the balance. For example, **Assignment Tax Unit Year to Date** indicates the balance value is for a given employee's assignment within a particular Tax Reporting Unit, for the Year-to-Date time period. Balance dimensions are predefined.

Contexts

Contexts are a means of restricting the run results that are included in a balance value. Contexts are normally predefined by either Global Payroll or localizations. Contexts are specified as part of the dimension.

Defined Balance

The defined balance is the name used to identify the combination of the Balance Type and the Balance Dimension (for example, REGULAR_ASG_TU_YTD). Whenever a balance value is obtained, it's from a defined balance.

Balance Model: How It All Works Together

A defined balance value is the value for an individual on a particular date for a specific balance type. Each defined balance holds a value for an individual at one of the following levels of the employment model:

- Payroll Relationship
- Assignment

The date is determined by the balance dimension, such as year-to-date. Balance dimensions may also use contexts, which are entities that require values for a particular balance value. For example, the defined balance **Regular Earnings Relationship Tax Unit Area1 Year to Date** is an association of the **Regular Earnings** balance type with the **Payroll Relationship Tax Unit Area1 Year to Date** balance dimension. The contexts associated with this dimension are **Payroll Relationship**, **Tax Reporting Unit**, and **Province (Area1)**.

A balance feed is an association between a balance type and an element input value. For example, if you specify that the salary pay value feeds the **Regular Earnings** balance, an employee's salary run result feeds all **Regular Earnings** defined balances. This is provided that the contexts associated with that run result match the contexts used by the defined balance.

Balance Initializations

Overview

When migrating payroll data from another application into cloud payroll, you might have to set the initial balance values. To do this, you can use HCM Data Loader to load balance values into batch views.

After loading the balance values, you submit the Load Initial Balances process. The process validates the batch data, and then processes the batch to load the balance data. It then creates balance adjustments to set the required balance values.

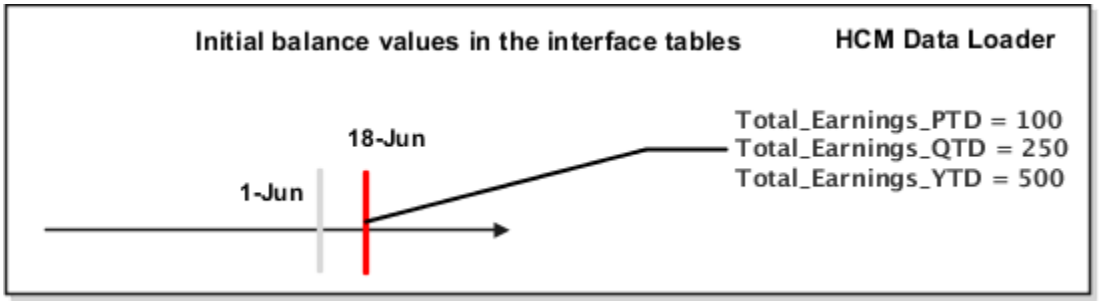
For each balance to initialize, the Load Initial Balances process defines these elements automatically with Balance Initialization classification. And adds them to the balance as balance feeds.

> **Note:** You can't initialize balances after the payroll is run for the employee. In such a case, you have to roll back the payroll run or do a balance adjustment.

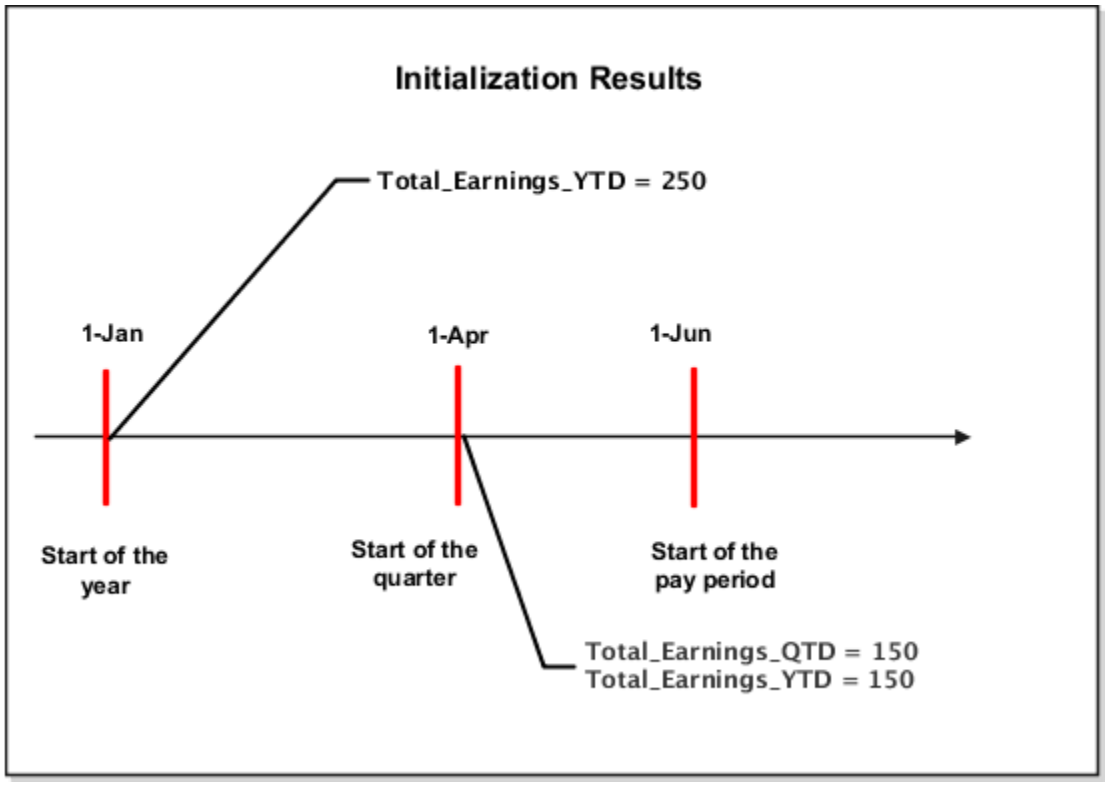
Example

In the middle of the year, your company hired a group of employees as part of an acquisition. The original company already paid them their earnings and withheld their taxes. In your company's ongoing payroll, you can initialize these employees' payroll balances for that year. This way, you ensure that all further statutory reports are accurate for the entire year and not just for the period they worked in your company.

As this image shows, you can initialize the total earnings balances for period-to-date, quarter-to-date, and year-to-date as of 18-Jun.



This image shows the initializing results as of 18-Jun.



Use the InitializeBalanceBatchHeader.dat file to create the VisionBatch batch.

```
METADATA|InitializeBalanceBatchHeader|LegislativeDataGroupName|BatchName|UploadDate
MERGE|InitializeBalanceBatchHeader|Vision Corporation US LDG|VisionBatch|2018/06/18|
```

In the InitializeBalanceBatchLine.dat file you initialize these three balance values for the total earnings balance.

Balance	Value
Total_Earnings_PTD	100
Total_Earnings_QTD	250
Total_Earnings_YTD	500

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|UploadDate|
PayrollRelationshipNumber|PayrollName|BalanceName|DimensionName|Value
MERGE|InitializeBalanceBatchLine|Vision Corporation US LDG|VisionBatch|1|2018/06/18|10001|Monthly Payroll|
Total Earnings|Core Relationship Period to Date|100
MERGE|InitializeBalanceBatchLine|Vision Corporation US LDG|VisionBatch|2|2018/06/18|10001|Monthly Payroll|
Total Earnings|Core Relationship Quarter to Date|250
MERGE|InitializeBalanceBatchLine|Vision Corporation US LDG|VisionBatch|3|2018/06/18|10001|Monthly Payroll|
Total Earnings|Core Relationship Year to Date|500
```

When you run the Load Initial Balances process, it sets the values for each balance relative to the upload date. For example, for an upload date of June 18, the balance values include all payment dates up to and including June 18. The process creates date-effective balance entries (adjustments) to ensure the legislative balances are correct.

As this table shows, the balance initialization process creates adjustments on the first day of the time period relevant to each dimension.

Adjustment Date	Adjusted Balances	Adjustment Value
1 June (Start of the pay period)	• Total_Earnings_PTD • Total_Earnings_QTD • Total_Earnings_YTD	100
1 April (Start of the quarter)	• Total_Earnings_QTD • Total_Earnings_YTD	150
1 Jan (Start of the year)	Total_Earnings_YTD	250

Note: In this example, the three dimensions (PTD, QTD and YTD) were initialized. However, you can simply initialize the YTD dimension by only including the YTD batch line on January 1 for 500.It is important to note that when you initialize balance values, (for example, Regular Salary) it will not feed other balances (for example, Gross Earnings). In the normal course of a payroll run, the application will maintain itself and this is not an issue. However, run results created from initial balance uploads only feed the specific balance for which they are loaded.

Considerations and Prerequisites for Balance Batches

When you create the batch header and lines, consider these aspects:

- Create separate balance initialization batches for each legislative data group (LDG).
- Although the HCM Data Loader upload is multithreaded, the subsequent balance processing is not. Therefore it's suggested that you create multiple batches so that the Load Initial Balances process may process concurrently.
- Within the same batch, ensure that you include lines for every balance to be initialized for a person. You can't split batch lines for the same person across multiple batches.
- The date you specify on the batch header record applies to all lines, unless you enter an override date at the line level. The date at the line level must be on or before the date on the header.
- You can't initialize balances once you have run any payroll processes.
- When initializing involuntary deduction balances, you must create the involuntary deduction card before processing the balance initialization.

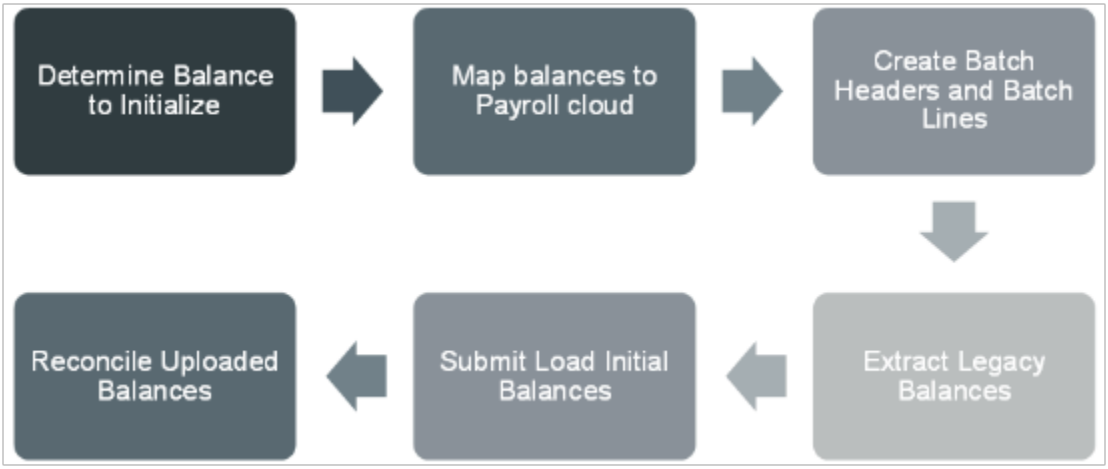
Related Topics

- Steps to Initialize Balances
- Example of Loading Initial Balances

Steps to Initialize Balances

You can use HCM Data Loader to load the initial balance values before you process the employee in a payroll. Before you start initializing the balances, group your employees into batches.

This image shows the steps involved in initializing balances.



Here are the steps that you do to initialize balances.

1. As the first step, you identify the balances to initialize. You can initialize legislative and user-defined balances. Each of the balances has its own initialization requirements.
2. Map the balances into the Global Payroll cloud. Remember, one legacy balance can feed one or more cloud balances. Also, many legacy balances can feed one cloud balance.
3. Extract legacy balances to load into the payroll cloud.
4. Load balance values to PAY_BAL_BATCH_HEADERS and PAY_BAL_BATCH_LINES interface tables by using these two separate HDL files.
    - InitializeBalanceBatchHeader.dat
    - InitializeBalanceBatchLine.dat
5. On the Home page, click the **Payroll Flow Patterns** quick action under the My Clients group tab. On the Payroll Flow Patterns page, search for and select the **Load Initial Balances** process to validate and transfer data from the batches into Global Payroll. You can also use the process to roll back the batch if you need to correct and reprocess errors. This process creates date-effective balance entries, or adjustments to ensure that the balances are correct from the upload date.
6. Check the balance load results to identify and correct any errors in the balances. Use the View Person Process Results and View Payroll Process Results tasks from the Payroll Calculation work area to reconcile the loaded balances.

Related Topics

- Example of Loading Initial Balances
- Overview of Balance Initialization

Example of Loading Initial Balances

In this example, you create balance initialization records to initialize an employee's federal and state-level balances.

When initializing year-to-date and quarter-to-date balances, you also initialize any related balances. For example, when initializing FIT Withheld balance, related balances such as FIT Exempt and FIT Gross too are initialized.

Load the Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the VisionBatch batch.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|VisionBatch|2015/02/05|PM US Sun Power
```

Load Batch Lines Data

Load these details to initialize the YTD salary balance.

Data Column	Value
Line Sequence	1
Balance	PM US Sun Power Salary
Dimension	Assignment Tax Unit Year to Date
Balance Value	6000.00

Load these details to initialize the QTD salary balance.

Data Column	Value
Line Sequence	2
Balance	PM US Sun Power Salary
Dimension	Assignment Tax Unit Quarter to Date
Balance Value	6000.00

Load these details to initialize the YTD FIT balance.

Data Column	Value
Line Sequence	3
Balance	FIT Withheld
Dimension	Relationship Tax Unit Year to Date
Balance Value	954.06

Load these details to initialize the QTD FIT balance.

Data Column	Value
Line Sequence	4
Balance	FIT Withheld
Dimension	Relationship Tax Unit Quarter to Date
Balance Value	954.06

Load these details to initialize the YTD Medicare balance.

Data Column	Value
Line Sequence	5
Balance	Medicare Employee Withheld
Dimension	Relationship Tax Unit Year to Date
Balance Value	87.00

Load these details to initialize the QTD Medicare balance.

Data Column	Value
Line Sequence	6

Data Column	Value
Balance	Medicare Employee Withheld
Dimension	Relationship Tax Unit Quarter to Date
Balance Value	87.00

Load these details to initialize the YTD SIT balance.

Data Column	Value
Line Sequence	7
Balance	SIT Withheld
Dimension	Relationship Tax Unit,State Year to Date
Balance Value	264.00
Area One	6 (Colorado)

Load these details to initialize the QTD SIT balance.

Data Column	Value
Line Sequence	8
Balance	SIT Withheld
Dimension	Relationship Tax Unit State Year to Date
Balance Value	264.00
Area One	6 (Colorado)

In the InitializeBalanceBatchLine.dat file, you initialize the YTD and QTD values for these balances.

- Salary
- FIT
- Medicare

- SIT

```
METADATA|InitializeBalanceBatchLine|BatchName|LineSequence|PayrollName|PayrollRelationshipNumber|TermNumber|
AssignmentNumber|BalanceName|DimensionName|TaxUnitName|AreaOne|Value
MERGE|InitializeBalanceBatchLine|VisionBatch|1|PM US Sun Power Weekly|955160008191951|ET955160008191951|
E955160008191951|PM US Sun Power Salary|Assignment Tax Unit Year to Date|PM US Sun Power Inc||6000
MERGE|InitializeBalanceBatchLine|VisionBatch|2|PM US Sun Power Weekly|955160008191951|ET955160008191951|
E955160008191951|PM US Sun Power Salary|Assignment Tax Unit Quarter to Date|PM US Sun Power Inc||6000
MERGE|InitializeBalanceBatchLine|VisionBatch|3|PM US Sun Power Weekly|955160008191951|ET955160008191951|
E955160008191951|FIT Withheld|Relationship Tax Unit Year to Date|PM US Sun Power Inc||954.06
MERGE|InitializeBalanceBatchLine|VisionBatch|4|PM US Sun Power Weekly|955160008191951|ET955160008191951|
E955160008191951|FIT Withheld|Relationship Tax Unit Quarter to Date|PM US Sun Power Inc||954.06
MERGE|InitializeBalanceBatchLine|VisionBatch|5|PM US Sun Power Weekly|955160008191951|ET955160008191951|
E955160008191951|Medicare Employee Withheld|Relationship Tax Unit Year to Date|PM US Sun Power Inc||87
MERGE|InitializeBalanceBatchLine|VisionBatch|6|PM US Sun Power Weekly|955160008191951|ET955160008191951|
E955160008191951|Medicare Employee Withheld|Relationship Tax Unit Quarter to Date|PM US Sun Power Inc||87
MERGE|InitializeBalanceBatchLine|VisionBatch|7|PM US Sun Power Weekly|955160008191951|ET955160008191951|
E955160008191951|SIT Withheld|Relationship Tax Unit,State Year to Date|PM US Sun Power Inc|6|264.00
MERGE|InitializeBalanceBatchLine|VisionBatch|8|PM US Sun Power Weekly|955160008191951|ET955160008191951|
E955160008191951|SIT Withheld|Relationship Tax Unit,State Quarter to Date|PM US Sun Power Inc|6|264.00
```

Related Topics

- Overview of Balance Initialization
- Steps to Initialize Balances

Balance Adjustments

Overview

For a given worker, you can use HCM Data Loader to adjust payroll balances as of a certain date. The balance adjustment process creates a run result that adjusts one or more balances depending upon the balance fed by the element input value.

You can decide whether to cost the balance adjustment results or not. Additionally, you can also decide whether you want the adjustment results to be paid or not. You can pay the employee when you run the next pre-payment process that includes these adjustment run results. Further, the process assigns the results to the appropriate payment method as per the worker's payment method setup.

Corrections to Balance Initialization

Recollect the scenario where a group of employees were hired through an acquisition in the middle of a year. The original company already paid them their earnings and withheld their taxes. In your company's ongoing payroll, you initialize these employees' payroll balances for that year. As you can't initialize balances after the payroll is run for the employee, you adjust the balances for them.

These scenarios describe how you can adjust balances after the first payroll run.

- The FIT Withheld balance for Andrew was initialized 100 less than what it should have been. The Payroll Manager makes a balance adjustment of 100 for Andrew using the payroll element that feeds the FIT Withheld balance.
- Jane Reifer was omitted during the balance initialization process and none of her balances were loaded. The Payroll Manager can adjust Jane's omitted balances.

Third-Party Gross to Net calculations

There are several kinds of earnings that are managed by a third-party. For example, a company could use a third-party to handle disability payments, stock option payouts, moving and relocation expenses. In these cases, the employer still does the reporting of wages or taxes. Also, they adjust the balances by importing the incremental values, which the third-party calculates and withholds, to the application. The company can cost these balance adjustments as the third-party has already made payments to the employee.

Let's consider this example. Prasad is on a long term disability leave and his employer uses a third-party to manage his disability leave payments.

The third-party does these tasks.

- Calculates Prasad's gross disability earnings as 5000 for the pay period.
- Withholds FIT, SS and Medicare taxes as 1000, 500 and 250, respectively.
- Pays him a net amount of 3250.

The Payroll Manager imports these values as balance adjustments for the balances.

Retrospective Changes Not Covered by RetroPay

Typically, you use the retroactive pay process to handle retroactive changes to data that impacts payroll calculation results. However, not all elements support Retro Pay. For example, tax deductions aren't enabled for retroactive pay. In such cases, for a retroactive change that impacts tax calculations, you can reflect the changes through balance adjustments.

Let's consider this example. John lives in California and pays his taxes there. He is on a biweekly payroll and is withheld 500 of the State Income Tax for the two weeks of earnings. In the 16 2018 Biweekly period, he worked in Texas, but didn't report the time on his time card. He should have been withheld SIT for 200 to Texas instead of the 500 to California.

In John's process results for the 16th payroll run, the payroll manager adjusts the existing SIT Withheld of 500 to California by doing these actions:

- Make a 200 adjustment to Work State Income Tax for the state of Texas.
- Negate the original run result for the amount 500.

Zeroing out Balances for Terminations

Here's an example where you zero out the balances for a terminated employee.

James Nance is terminated as of 9/15/2018 with a final close date of 9/30/2018. The payroll manager would like to zero out all arrears balances for James as of 9/30. Upon termination, the manager runs the deductions report and finds that James has these arrears balances as of 9/30.

- Car Loan Deduction 2000.00
- Student Loan Deduction 500.00

For the Arrears Adjustment element, the payroll manager loads -2000.00 and -500 as input values. These values zero out the Inception to Date arrears balances.

Related Topics

- Example of Loading Balance Adjustments
- Examples of Balance Exceptions

Steps to Adjust Balances

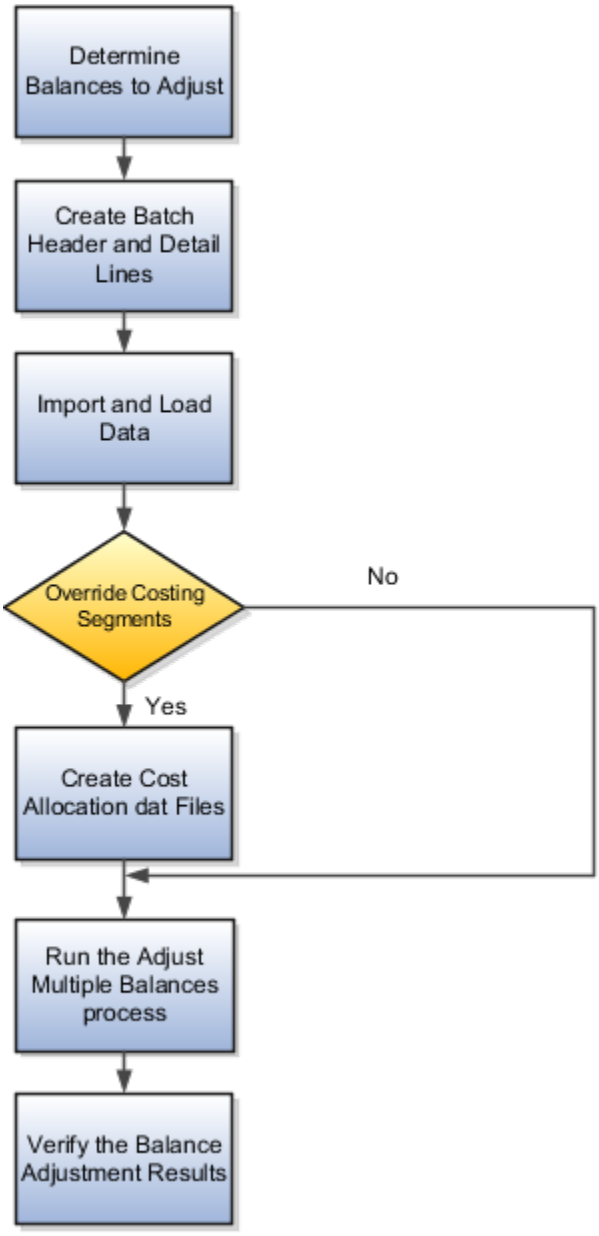
You can use HCM Data Loader to load balance adjustments in the batch mode.

Before You Begin

Before you adjust balances, do these tasks:

- Create the appropriate eligibility links for the elements that you're adjusting. For example, to adjust base elements, it must contain an eligibility link.
- Populate the Balance Date column if the balance dimension requires an entry.

This image shows the steps involved in adjusting balances.



Perform these steps to adjust your balances.

1. Determine the balances that require an adjustment.
2. Create the batch header and detail lines in the dat files. Use the balance adjustment header to adjust balances in bulk. The balance adjustment lines contain the individual adjustment lines for the balance adjustment group.
3. Import and load the data using the **HCM Data Load** process. This process creates a run result by adjusting one or more balances. And this adjustment depends upon the balance fed by the element input value.
4. Decide whether the balance adjustment results must be costed or not. If you want to cost your balance adjustments, you can create the cost allocation dat files.

> **Note:** Not all costed adjustments need the cost allocation files. Only those costed adjustments that need one or more costing segments to be overridden for the adjustment element entry need the allocation file

5. From Payroll Calculation work area, run the **Adjust Multiple Balances** process to create balance adjustment element entries for each line within the data set. In this process, you provide the batch name that you used in the HCM Data Loader file.
6. Verify the balance results of adjustments.

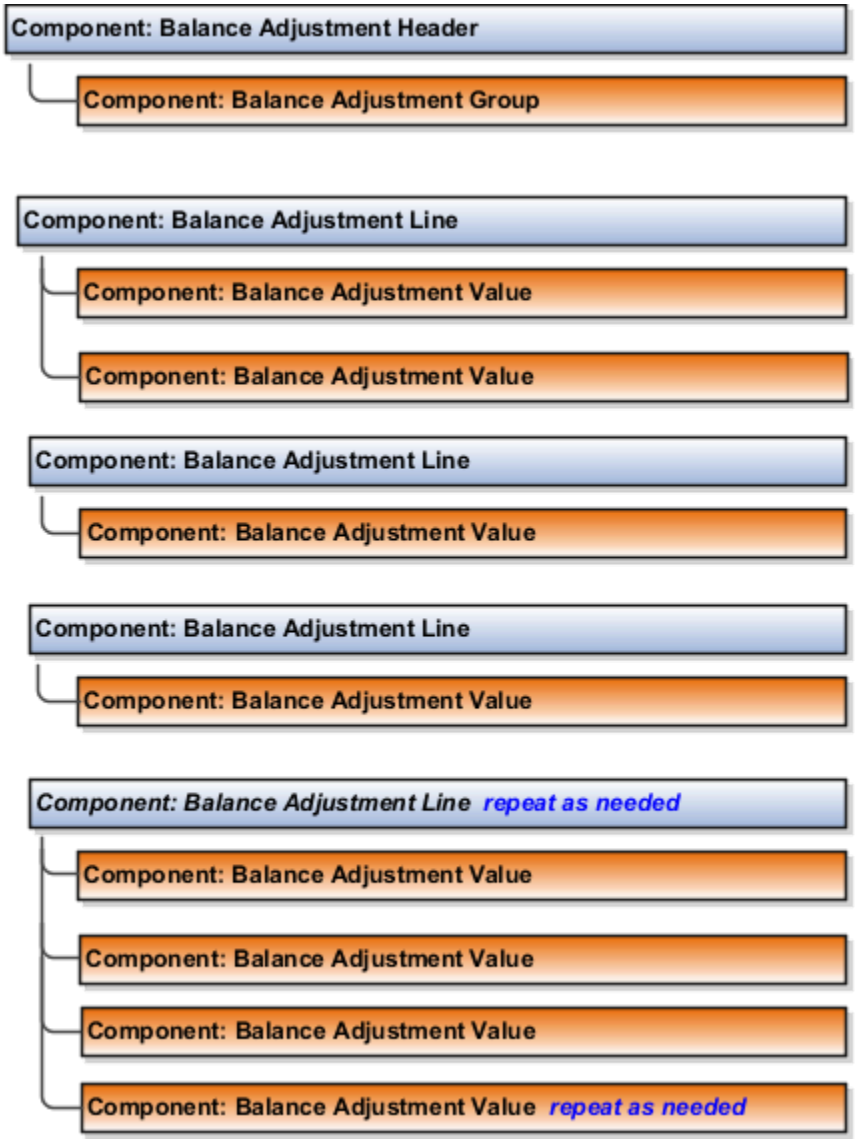
Related Topics

- Balance Adjustment Record Types
- Example of Loading Balance Adjustments
- Examples of Balance Exceptions

Balance Adjustment Record Types

Use the Balance Adjustment Header and the Balance Adjustment Line business objects to upload the balance adjustment details.

This image shows the hierarchy of components that are applicable to the batch data.



This table summarizes the various balance adjustment components:

Component	What It's
Balance Adjustment Header	The header for the batch process that's used when adjusting balances in bulk. Supply one batch header record for each group.
Balance Adjustment Group	A group of balance adjustments within the batch header that correspond to a unique combination of payroll, consolidation group, and effective date. Supply one group record for each batch header. At this time, you may only have one group record per batch.
Balance Adjustment Line	Individual adjustment lines for the balance adjustment group. Each line corresponds to a specific adjustment element entry for a worker and the associated contexts for the adjustment run result.
Balance Adjustment Value	Individual adjustment values for each balance adjustment line. Each value corresponds to a specific adjustment entry value for a worker and adjusts the balances it feeds.

Component	What It's

Related Topics

- Overview of Loading Balance Adjustments
- Steps to Adjust Balances
- Examples of Balance Exceptions

Example of Loading Balance Adjustments

Vision Corp wants to create balance adjustment records to adjust the FIT Withheld balances for two of their employees. Using HCM Data Loader, they want to increase the withheld balance by 10.

Load the Balance Adjustment Header

Use this BalanceAdjustmentHeader.dat file to create the batch header.

```
METADATA|BalanceAdjustmentHeader|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|US LDG|FIT Adjustments
METADATA|BalanceAdjustmentGroup|LegislativeDataGroupName|BatchName|EffectiveDate|PayrollName|
ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|US LDG|FIT Adjustments|2019/04/30|Vision US Weekly|Vision Weekly|N|N
```

Load the Balance Adjustment Line and Value

For Employee 1 with Assignment number E3263769, load these details:

Data Column	Value
Line Sequence	1
Effective Date	2010/04/30
Payroll	Vision US Weekly
Consolidation Group	Vision Weekly
Assignment Number	E3263769
Include adjustment in costing process	No
Federal Income Tax, Tax Calculated	10.00
Include adjustment in payment balance	No

For Employee 2 with Assignment number E5746169, load these details:

Data Column	Value
Line Sequence	2
Effective Date	2010-04-30
Payroll	Vision US Weekly
Consolidation Group	Vision Weekly
Assignment Number	E5746169
Include adjustment in costing process	No
Federal Income Tax, Tax Calculated	10.00
Include adjustment in payment balance	No

Use this BalanceAdjustmentLine.dat file to create the batch lines:

```
METADATA|BalanceAdjustmentLine|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|AssignmentNumber|ElementName|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|US LDG|FIT Adjustments|Vision US Weekly|Vision Weekly|2019/04/30|1|E263769|
Federal Income Tax|US Tax Reporting Unit
MERGE|BalanceAdjustmentLine|US LDG|FIT Adjustments|Vision US Weekly|Vision Weekly|2019/04/30|2|E746169|
Federal Income Tax|US Tax Reporting Unit
METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|InputValueName|EntryValue|ElementName
MERGE|BalanceAdjustmentValue|US LDG|FIT Adjustments|Vision US Weekly|Vision Weekly|2019/04/30|1|Tax
 Calculated|10|Federal Income Tax
MERGE|BalanceAdjustmentValue|US LDG|FIT Adjustments|Vision US Weekly|Vision Weekly|2019/04/30|2|Tax
 Calculated|10|Federal Income Tax
```

If you wish, you can also specify a calculation component. For example, customers in UK can specify a PAYE Component for the balance adjustment and customers in Canada can specify a Payroll Account Number (PAN). Here's a sample BalanceAdjustmentLine.dat file:

```
METADATA|BalanceAdjustmentLine|BatchName|LegislativeDataGroupName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|AssignmentNumber|ElementName|TaxReportingUnitName|BalanceDate|
ThirdPartyPayeeName|TimeDefinitionName|CalcBreakdownId|CalculationComponent
MERGE|BalanceAdjustmentLine|HDL_Batch1|LDG UK1|Monthly PD|LDG CS UK1|2019/06/01|31|E955160008191984|
Sample_Uk_Element|LE 2T|2019/06/01||UKBT_TD_WEEKLY||LE 2T
```

Load Balance Adjustment Costing

BalanceAdjustmentHeader.dat

```
METADATA|BalanceAdjustmentHeader|BatchName|LegislativeDataGroupName
MERGE|BalanceAdjustmentHeader|WK_BA_501_03012021|WK_US_LDG

METADATA|BalanceAdjustmentGroup|EffectiveDate|LegislativeDataGroupName|BatchName|PayrollName|
ConsolidationSetName|BalanceAdjCostFlag
```

```
MERGE|BalanceAdjustmentGroup|2021/03/01|WK_US_LDG|WK_BA_501_03012021|WK_Monthly|WK Consolidation Group|Y
```

WK_BA_501_03012021_header.zip UCMFA00151313 Success Success

BalanceAdjustmentLine.dat

```
METADATA|BalanceAdjustmentLine|BalanceDate|BatchLineSequence|LegislativeDataGroupName|EffectiveDate|
BatchName|PayrollName|AssignmentNumber|ElementName|TaxReportingUnitName|ConsolidationSetName
MERGE|BalanceAdjustmentLine|2021/03/01|1|WK_US_LDG|2021/03/01|WK_BA_501_03012021|WK_Monthly|E501|Federal
 Income Tax|WK_US_LE|WK Consolidation Group

METADATA|BalanceAdjustmentValue|EntryValue|InputValueName|LegislativeDataGroupName|BatchName|
BatchLineSequence|PayrollName|EffectiveDate|ConsolidationSetName|ElementName
MERGE|BalanceAdjustmentValue|-211|Gross|WK_US_LDG|WK_BA_501_03012021|1|WK_Monthly|2021/03/01|WK
 Consolidation Group|Federal Income Tax
MERGE|BalanceAdjustmentValue|-211|Tax Calculated|WK_US_LDG|WK_BA_501_03012021|1|WK_Monthly|2021/03/01|WK
 Consolidation Group|Federal Income Tax
```

WK_BA_890217_03012021_lines.zip UCMFA00151325 Success Success

CostAllocationV3.dat

```
METADATA|CostAllocationV3|BatchName|EffectiveStartDate|Consolidationsetname|BalAdjGrpEffectiveDate|
Batchlinesequence|PayrollName|AssignmentNumber|LegislativeDataGroupName|SourceType
MERGE|CostAllocationV3|WK_BA_890217_03012021|2021/03/01|WK_Consolidation_Group|2021/03/01|1|WK_Payroll|
E890217|WK_LDG|BA
```

WK_BA_890217_03012021_cost_alloc.zip UCMFA00485453 Success Success

CostAllocationAccountV3.dat

```
METADATA|CostAllocationAccountV3|SourceType|PayrollName|BatchName|EffectiveDate|BalAdjGrpEffectiveDate|
Batchlinesequence|LegislativeDataGroupName|Consolidationsetname|SourceSubType|Proportion|SubTypeSequence|
ConcatenatedSegment|Segment1|Segment2|Segment3
MERGE|CostAllocationAccountV3|BA|WK_Payroll|WK_BA_890217_03012021|2021/03/01|2021/03/01|1|WK_LDG|
WK_Consolidation_Group|COST|1|1|000.1234.1515|000|1234|1515
```

WK_BA_890217_03012021_CostAllocAcct.zip UCMFA00485503 Success Success

Submit process

Payroll Checklist > Adjust Multiple Balances

> **Note:** Ensure you run the Adjust Multiple Balances process, **after** ALL 4 loads mentioned above, **in sequence**, to see the costing in the user interface.

Related Topics

- Overview of Loading Balance Adjustments
- Steps to Adjust Balances
- Balance Adjustment Record Types

18 Loading Payment Methods

Examples of Loading Banks

A bank is a financial institution that can have multiple branches. External bank accounts are associated with bank branches. Bank, Bank Branch, and External Bank Account are business objects whose details are stored in non-HCM tables.

However, you can load them using HCM Data Loader. This topic describes aspects of the Bank object that you must understand to load banks successfully.

Loading Banks

You load bank data in a Bank.dat file for processing by HCM Data Loader. To provide a unique reference to a bank when you create it, you must supply either the **Bank Name** or the **Bank Number** attribute. You must also supply the **Country Code** attribute.

You are recommended to use source keys when creating banks. This makes it easier to reference the bank when creating branches or external bank accounts.

This example Bank.dat file creates a Bank object using source keys. Both the Bank Name and Bank Number attributes are provided.

```
METADATA|SourceSystemOwner|SourceSystemId|Bank|BankName|BankNumber|CountryCode
MERGE|Bank|VISION|CA10001A|Vision Bank|100001A|CA
```

This example Bank.dat file creates a Bank object using user keys. Both the Bank Name and Bank Number attributes are provided.

```
METADATA|Bank|BankName|BankNumber|CountryCode
MERGE|Bank|Vision Bank|100001A|CA
```

> **Note:** When you create a bank using HCM Data Loader, it's created with an institution type of **Bank**. You can only maintain and reference banks with this institution type. If you create the bank using any other method, the HCM Data Loader can't find the bank when you try creating a bank account later.

Deleting Banks

You can't delete Bank objects using HCM Data Loader. However, you can inactivate banks that are either no longer required or were entered in error. This feature, which isn't available on the Manage Banks pages, enables you to maintain an audit trail.

This example Bank.dat file provides an end date for a Bank object to inactivate it. To identify an existing bank uniquely, you can use the **Country Code** and either the **Bank Name** or the **Bank Number**. Alternatively, supply the source key to identify the bank to be inactivated.

> **Note:** When you inactivate a bank, any updates to other attributes are ignored.

```
METADATA|Bank|BankNumber|CountryCode|EndDate
```

```
MERGE|Bank|100001A|CA|2016/03/01
```

Related Topics

- Examples of Loading Bank Branches
- Examples of Loading External Bank Accounts

Examples of Loading Bank Branches

A bank is a financial institution that can have multiple branches. External bank accounts are associated with bank branches. Bank, Bank Branch, and External Bank Account are business objects whose details are stored in non-HCM tables.

However, you can load them using HCM Data Loader. This topic describes aspects of the Bank Branch object that you must understand to load bank branches successfully.

Loading Bank Branches

Bank branch data can be supplied in a BankBranch.dat file for processing by HCM Data Loader. The **Bank Branch Number** attribute, for which the field name on the Manage Bank Branches page varies by country, uniquely identifies the bank branch. It may be a required attribute, depending on country-specific validations.

You can supply one of the following combinations of attributes to provide a unique reference to the bank branch and the associated bank:

- **Bank Branch Name**, **Country Code**, and either **Bank Name** or **Bank Number**
- **Bank Branch Number**, **Country Code**, and either **Bank Name** or **Bank Number**

Alternatively, supply a source key to identify the branch and the bank it belongs to.

This example BankBranch.dat file creates a Bank Branch using the **Bank Branch Number**, **Bank Name**, and **Country Code** user keys.

```
METADATA|BankBranch|BankBranchNumber|BankBranchName|BankName|CountryCode|EftSwiftCode
MERGE|BankBranch|111111A|Toronto 3|Vision Bank|CA|12345678
```

This example BankBranch.dat file creates a Bank Branch, using source keys to identify the branch and bank.

```
METADATA|BankBranch|SourceSystemOwner|SourceSystemId|BankId(SourceSystemId)|BankBranchNumber|BankBranchName|
BankName|CountryCode|EftSwiftCode
MERGE|BankBranch|VISION|111111A|CA10001A|111111A|Toronto|CA|12345678
```

Note: When you create a bank branch using HCM Data Loader, it's created with an institution type of **Bank Branch**. You can only maintain and reference bank branches with this institution type. If you create the bank branch using any other method, the HCM Data Loader can't find the branch when you try creating a bank account later.

Deleting Bank Branches

You can't delete Bank Branch objects using HCM Data Loader. However, you can inactivate bank branches that are no longer required or were entered in error. This feature, which isn't available on the Manage Bank Branches pages, enables you to maintain an audit trail.

Note: When you inactivate a bank, any updates to other attributes are ignored.

This example BankBranch.dat file provides an end date to inactivate a bank branch.

```
METADATA|BankBranch|BankBranchNumber|BankName|CountryCode|EndDate
MERGE|BankBranch|111111A|Vision Bank|CA|2015/04/01
```

Related Topics

- Examples of Loading Banks
- Examples of Loading External Bank Accounts

Examples of Loading External Bank Accounts

An external bank account record holds the details of a bank account at a bank branch. Bank accounts are used by payment methods to make payments to a person.

Bank, Bank Branch, and External Bank Account are business objects whose details are stored in non-HCM tables. However, you can load them using HCM Data Loader. This topic describes aspects of the External Bank Account object that you must understand to load external bank accounts successfully.

Bank and Bank Branch

Before you can create an External Bank Account object, you must create both the bank and the bank branch where the bank account is held.

Note: When you create a bank or a bank branch using HCM Data Loader, they're created with the institution types of **Bank** and **Bank Branch** respectively. You can only maintain and reference banks and bank branches with these institution types. If you create the bank or the branch using any other method, the HCM Data Loader can't find them when you try creating a bank account later.

Loading External Bank Accounts

You supply external bank account data in the ExternalBankAccount.dat file for processing by HCM Data Loader. Multiple user keys are supported for referencing the external bank account. For details of these user keys, view the External Bank Account object on the Business Object Details page, using the View Business Objects task.

The External Bank Account Owner component identifies a person or third party to whom the account belongs. A bank account can be associated with multiple owners, of whom one must be the primary owner. Use the discriminator ExternalBankAccountOwner to load External Bank Account Owner components. Set the **PrimaryFlag** attribute of the relevant component to **Y** to identify the primary owner. Multiple user keys are supported for referencing the external bank account owner. For details of these user keys, view the External Bank Account Owner component on the Business Object Details page.

Note: You can't load external bank accounts for pending workers.

It is recommended that you use source keys to identify the external bank account and to reference the bank and bank branch the account is for. To support the various localization requirements there are a large number of user keys provided, supplying source keys removes that complexity.

This ExternalBankAccount.dat example uses source keys to identify the bank, bank branch, external bank account, and the external bank account owner.

```
METADATA|ExternalBankAccount|SourceSystemOwner|SourceSystemId|BankId(SourceSystemId)|
BankBranchId(SourceSystemId)|CountryCode|CurrencyCode|AccountNumber|AccountName|AccountType
MERGE|ExternalBankAccount|VISION|12345678_CA|CA10001A|111111A|CA|CAD|12345678|J and P Smith|SAVINGS

METADATA|ExternalBankAccountOwner|SourceSystemOwner|SourceSystemId|ExternalBankAccountId(SourceSystemId)|
PersonId(SourceSystemId)|PrimaryFlag
MERGE|ExternalBankAccountOwner|VISION|12345678_CA_121011|12345678_CA|SK121011|Y
```

This example ExternalBankAccount.dat file uses user keys to create an External Bank Account and associate it with two owners, one of whom is the primary owner.

```
METADATA|ExternalBankAccount|BankNumber|BankBranchNumber|CountryCode|AccountNumber|IBAN|AccountName|
CurrencyCode
MERGE|ExternalBankAccount|100001A|111111A|CA|12345678|CA23 ANBK 3350 1234 5678 20|J and P Smith|USD
METADATA|ExternalBankAccountOwner|BankNumber|BankBranchNumber|CountryCode|AccountNumber|CurrencyCode|
PersonNumber|PrimaryFlag
MERGE|ExternalBankAccountOwner|100001A|111111A|CA|12345678|USD|121011|Y
MERGE|ExternalBankAccountOwner|100001A|111111A|CA|12345678|USD|126231|N
```

Inactivating External Bank Accounts

You can delete an External Bank Account object, and recreate it if necessary, using HCM Data Loader. Alternatively, you can make the bank account inactive. This example ExternalBankAccount.dat file makes an external bank account inactive by including the **InactiveFlag** attribute and setting it to **Y**.

Note: When you inactivate a bank account, any updates to other attributes are ignored.

```
METADATA|ExternalBankAccount|BankNumber|BranchNumber|CountryCode|CurrencyCode|AccountNumber|InactiveFlag
MERGE|ExternalBankAccount|100001A|111111A|CA|USD|12345678|Y
```

Related Topics

- Examples of Loading Banks
- Examples of Loading Bank Branches

Organization Payment Methods

Overview of Loading Organization Payment Methods

An organization payment method defines the payment methods for a legislative data group. Payment methods specify a payment type, such as check or direct deposit, a source bank account, and other details.

Using HCM Data Loader, load at least one organization payment method for each combination of legislative data group, payment type, and currency.

To load an organization payment method, you define at least one payment source. If you load multiple payment sources, then you can create payment rules. The payment rules determine which payment source to use for each TRU and also validate or process the distribution of payments.

Create payment method rules with effective start dates on or before the start dates of other objects that refer to your payment method rule.

This figure shows the steps to create an Organization Payment Method.

1. Create an Organization Payment Method.
2. Load the payment sources to associate bank accounts and other sources of funds with it.
3. Define payment rules.

Related Topics

- Example of Loading Organization Payment Methods
- Organization Payment Methods Overview
- How Payment Methods and Payroll Definitions Work Together
- How do I import organization data?

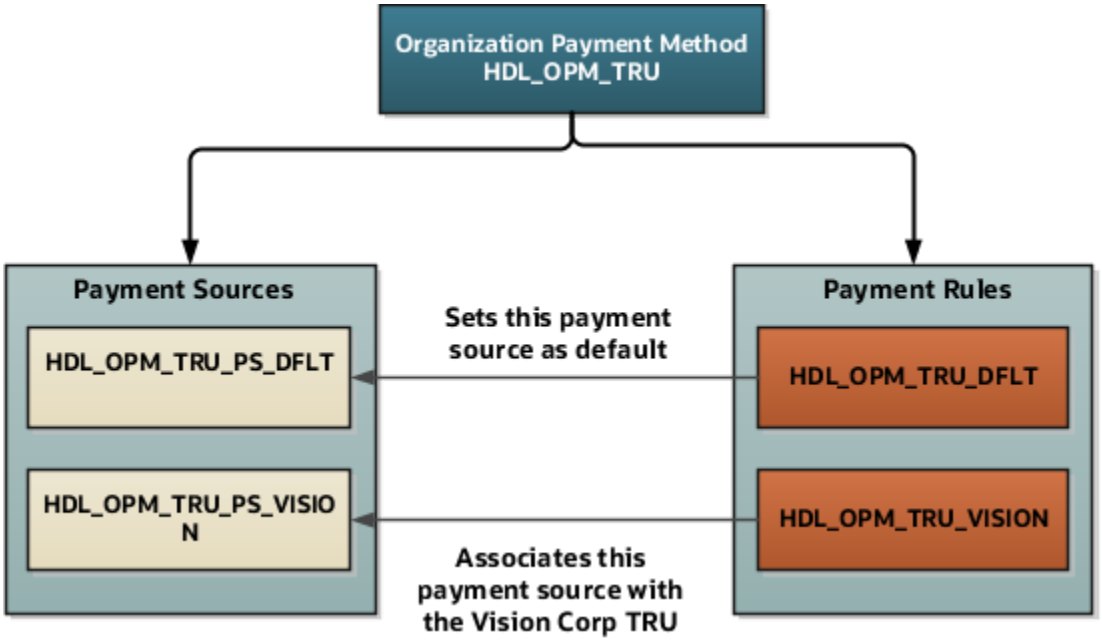
Example of Loading Organization Payment Methods

In this example, you create an Organization Payment Method and define payment sources and payment rules.

You do these steps:

1. Create an Organization Payment Method **HDL_OPM_TRU** for the **Vision Corporation US LDG** LDG.
2. Load the two payment sources to associate bank accounts and other sources of funds with it.
3. Define the payment rules to determine the appropriate payment source based on tax reporting unit.

As this image shows, for your organization payment method, you create the payment sources and rules.



Create an Organization Payment Method

These payment method lines create an organization payment method **HDL_OPM_TRU**:

```
METADATA|OrganizationPaymentMethod|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
EffectiveStartDate|OrganizationPaymentMethodName|OrganizationPaymentMethodCode|CurrencyCode|PaymentTypeCode
MERGE|OrganizationPaymentMethod|VISION|HDL_OPM_TRU|Vision Corporation US LDG|2018/01/01|HDL Organization
 Payment Method with TRU rule|HDL_ORG_PAYMENT_METHOD_TRU|USD|EFT
```

Create Payment Sources

These DAT files to load the two payment sources:

Payment Source	Use To
HDL_OPM_TRU_PS_VISION	Pay workers or third-party people based in the Vision Corporation tax reporting unit.
HDL_OPM_TRU_PS_DFLT	Pay workers in all other tax reporting units. Set this payment source as the default payment source.

```
METADATA|PaymentSource|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
OrganizationPaymentMethodId(SourceSystemId)|PaymentSourceCode|PaymentSourceName|BankAccountName|
BankReferenceEFT|CompanyReferenceEFT

MERGE|PaymentSource|VISION|HDL_OPM_TRU_PS_DFLT|Vision Corporation US LDG|2018/01/01|HDL_OPM_TRU|HDL_DEFLT|
HDL Payment Source|Corporate Payroll Account-202|808123663|456102113

MERGE|PaymentSource|VISION|HDL_OPM_TRU_PS_VISION|Vision Corporation US LDG|2018/01/01|HDL_OPM_TRU|
HDL_VISION|HDL Payment Source Vision|Vision Nacha Account-202|234506332|567890123
```

Create Payment Rules

Finally, use the DAT file to load these two payment rules.

Payment Rule	What It Does
HDL_OPM_TRU_DFLT	Sets the payment source HDL_OPM_TRU_PS_DFLT as the default source.
HDL_OPM_TRU_VISION	Associates the payment source HDL_OPM_TRU_PS_VISION with the Vision Corporation TRU.

```
METADATA|PaymentMethodRule|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
PaymentSourceId(SourceSystemId)|DefaultFlag|TaxReportingUnitName

MERGE|PaymentMethodRule|VISION|HDL_OPM_TRU_DFLT|Vision Corporation US LDG|2018/01/01|HDL_OPM_TRU_PS_DFLT|Y|

MERGE|PaymentMethodRule|VISION|HDL_OPM_TRU_VISION|Vision Corporation US LDG|2018/01/01|
HDL_OPM_TRU_PS_VISION|N|Vision Corporation
```

Note: When updating these existing components, you can't change either the first effective start date or last effective end date:

- Organization payment method
- Payment source
- Payment method rule

Related Topics

- Organization Payment Methods Overview
- How Payment Methods and Payroll Definitions Work Together
- Payment Sources in Organization Payment Methods Setup Examples
- How do I import organization data?

Example of Skipping Prenotes

A prenote is an entry you must send a few days prior to the first live payroll credit issued. The purpose of a prenote is to validate routing number and account numbers of the receiving bank or credit union.

If prenote has already been performed on a particular bank account, then use the **Payroll Bank Account Prenote Status Update** process to skip the prenote process.

Essentially, this process creates a prenote record with a status of **Marked as complete** for a specific bank account. This action stops revalidating the bank account.

When you load a prenote file, the application populates the prenote date as 01/01/00. And sets the prenote status on the personal payment method as **Marked as complete**.

Let's consider this example. Vision Corp Company has its headquarters in US. The company subjects its employees that have elected to receive direct deposits to the prenote process. They have set up their organization payment method with **Prenotification Required** option as **Yes**.

Recently, the company moved their data into cloud. Now, the application automatically triggers the prenote process. This means that the company's employees will be most likely paid their first payment by check. But the company doesn't want such a type of payment to happen and wants to skip the prenote process. Using HCM Data Loader, you can skip the prenote process for external bank accounts.

Use the `PayBankAccountPrenote.dat` file for updating the prenote status for your employees' bank account.

```
METADATA|PayBankAccountPrenote|BankName|BankBranchName|BankAccountType|BankAccountNumber|BankCountryCode
MERGE|PayBankAccountPrenote|ABCBank|NewYorkCityBranch|Savings|123456789|US
```

Related Topics

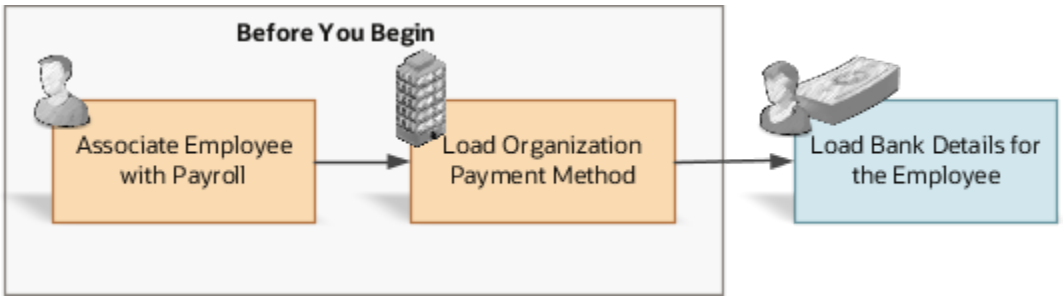
- Example of Loading Organization Payment Methods

Example of Loading Personal Payment Methods

Use HCM Data Loader to load payment method details for a person. Personal payment methods associate people to specific payment method, currency, and payment source.

In this example, Jane signed up for direct deposit of her pay check. Effective January 1, 2016, you create a payment method for her.

As this image shows, loading a personal payment method is a four step process.



Before creating a personal payment method, do these tasks:

- Associate Jane with a payroll.
- Load the organization payment method.
- Create the bank account the personal payment method will pay into.

As Jane's payment type is **Direct Deposit**, load these bank account details for her account:

Bank Account	Value
Account Number	1265896441
Bank Name	Bank of America
Routing Transit Number (Bank Branch Number)	122122125

Use the **PersonalPaymentMethod.dat** file to create a personal payment method. This file deposits only the first $10000 into the bank account.

```
METADATA|PersonalPaymentMethod|LegislativeDataGroupName|AssignmentNumber|PersonalPaymentMethodCode|
EffectiveStartDate|PaymentAmountType|Amount|ProcessingOrder|OrganizationPaymentMethodCode|Percentage|
BankName|BankBranchNumber|BankCountryCode|BankAccountNumber
MERGE|PersonalPaymentMethod|Vision Corporation US LDG|234299A|PPM3_HDL_NO_PO|2016/01/01|M|10000||PM US Sun
 Power Check||Bank of America|511000025|US|1265896441
```

Related Topics

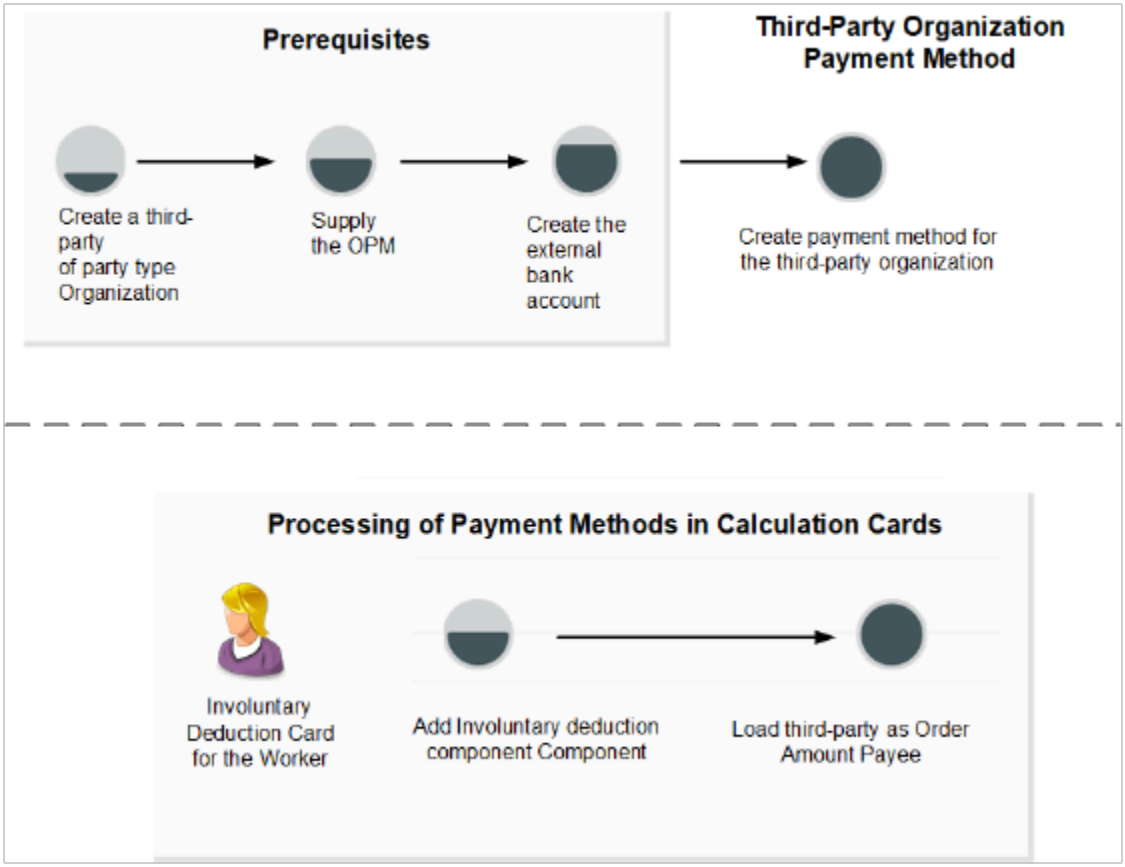
- Example of Loading Organization Payment Methods
- Examples of Loading External Bank Accounts
- How Payment Methods and Payroll Definitions Work Together
- Configure Payment Method Preferences
- How do I import organization data?

Example of Loading Third-Party Organization Payment Method

Create payment methods to make payments to external organizations, such as pension providers or professional bodies. These payment methods include details, such as the type of payment and the bank account details.

Example

This image shows how you associate a third-party organization representing a Texas Court to a payment method. This enables the payment of child support deductions that are deducted through the payroll to be paid to the court. Also, you add a calculation card to an employee for the involuntary deduction for the child support payment.



Before You Begin

Do these steps:

1. Create a third-party organization.
2. Load the Organization Payment Methods for your payroll.
3. Load the external bank account.

Load the External Bank Account

You can use the External Bank Account Owner component to identify the third-party to whom the account belongs to.

This example ExternalBankAccount.dat file creates an External Bank Account component and associates it with a third-party organization.

```
METADATA|ExternalBankAccount|BankNumber|BankBranchNumber|CountryCode|AccountNumber|IBAN|AccountName|
CurrencyCode
MERGE|ExternalBankAccount|100001A|111111A|CA|12345678|CA23 ANBK 3350 1234 5678 20|J and P Smith|USD
METADATA|ExternalBankAccountOwner|BankNumber|BankBranchNumber|CountryCode|AccountNumber|CurrencyCode|
ThirdPartyNumber|PrimaryFlag
MERGE|ExternalBankAccountOwner|100001A|111111A|CA|12345678|USD|68299|Y
```

Use this dat file to associate a payment method with the third-party organization HDL_Texas_ORG and organization payment method Direct Deposit. Additionally, as the payment type is Direct Deposit, you load these bank account details.

Bank Account Component	Value
Account Number	1265896441
Bank Name	Bank of America
Bank Branch Name	New York
Routing Transit Number	511000025
Account Type	Checking

```
METADATA|ThirdPartyOrganizationPaymentMethod|EffectiveStartDate|EffectiveEndDate|LegislativeDataGroupName|
TimeDefinitionCode|PartyNumber|BankName|BankBranchNumber|BankAccountNumber|BankCountryCode|
OrganizationPaymentMethodCode
MERGE|ThirdPartyOrganizationPaymentMethod|2016/01/01||USA LDG||68299|Bank of America|511000025|1265896441|
US|OPM_US_Nacha
```

Related Topics

- Example of Loading Organization Payment Methods
- Example of Loading Third-Party Personal Payment Method
- How Payment Methods and Payroll Definitions Work Together
- How do I import organization data?

Example of Loading Third-Party Personal Payment Method

Use HCM Data Loader to create payment methods to people who aren't on your payroll. When loading such payment methods, associate the third-party persons directly with the employee whose pay is subjected to deduction by supplying the assignment number.

A third-party person can receive payments from more than one employee too.

Example

John Smith is an employee at your organization. Mary Smith is his former spouse and receives a child-support payment for each payroll period that's deducted from John's salary. You want to set up payments for Mary.

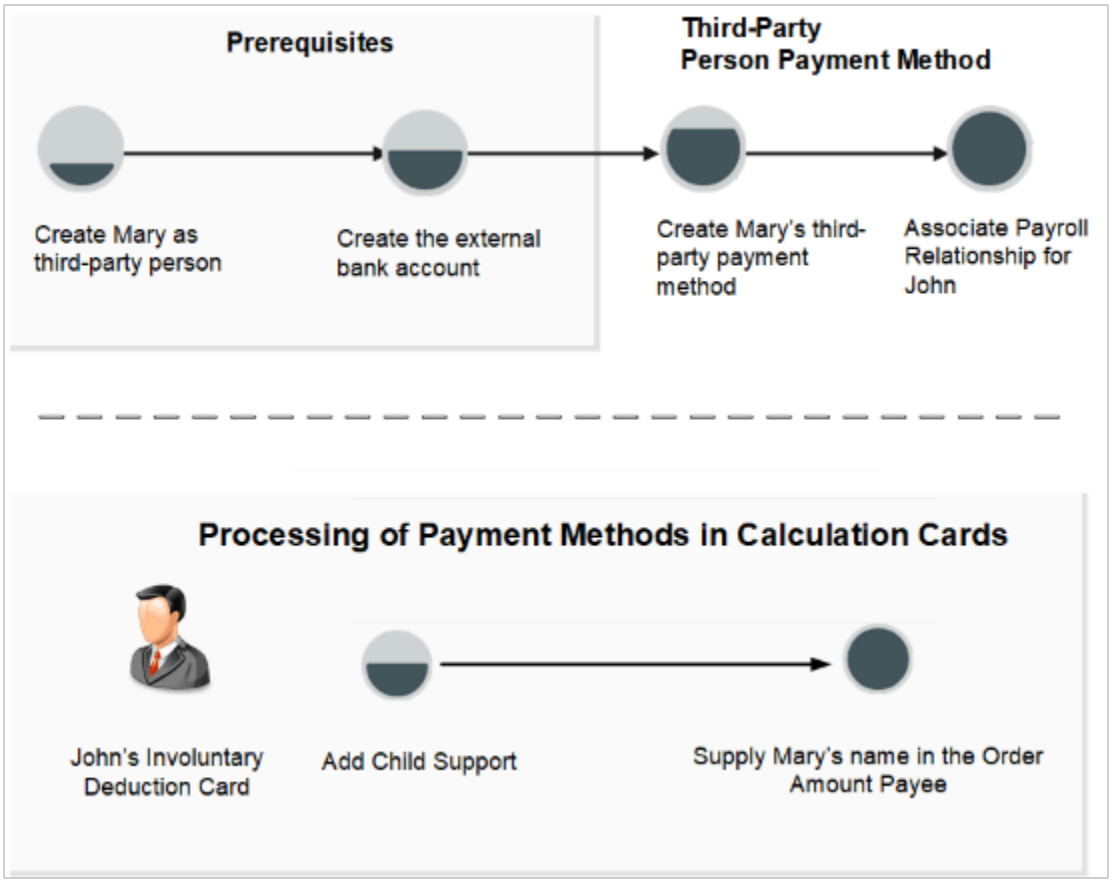
Before You Begin

Do these steps:

1. Create Mary as a third-party person.
2. Create the external bank account details.

Then, you create Mary's third-party payment method and associate the payroll relationship for John by supplying a reference to his assignment. Finally, add the child support to John's involuntary deduction calculation card and supply Mary's name as Order Amount Payee.

This figure shows how you can pay a third-party person using payment methods and calculation cards.



Use this example dat file to associate a payment method with the third-party person Mary Smith. The payment type for Mary is **Check**. The association with John's payroll relationship is achieved by supplying a reference to his assignment.

```
METADATA|ThirdPartyPersonalPaymentMethod|EffectiveStartDate| LegislativeDataGroupName|
OrganizationPaymentMethodCode|AssignmentNumber|PartyNumber|BankName|BankBranchNumber|BankCountryCode|
BankAccountNumber
MERGE|ThirdPartyPersonalPaymentMethod|2016/01/01|USA LDG|Vision US1|E95516|395077|HSBC|568793|US|12345678
```

Related Topics

- Example of Loading Organization Payment Methods
- Example of Loading Third-Party Organization Payment Method
- How Payment Methods and Payroll Definitions Work Together
- Third Parties Overview
- How do I import organization data?

19 Loading Payroll Costing

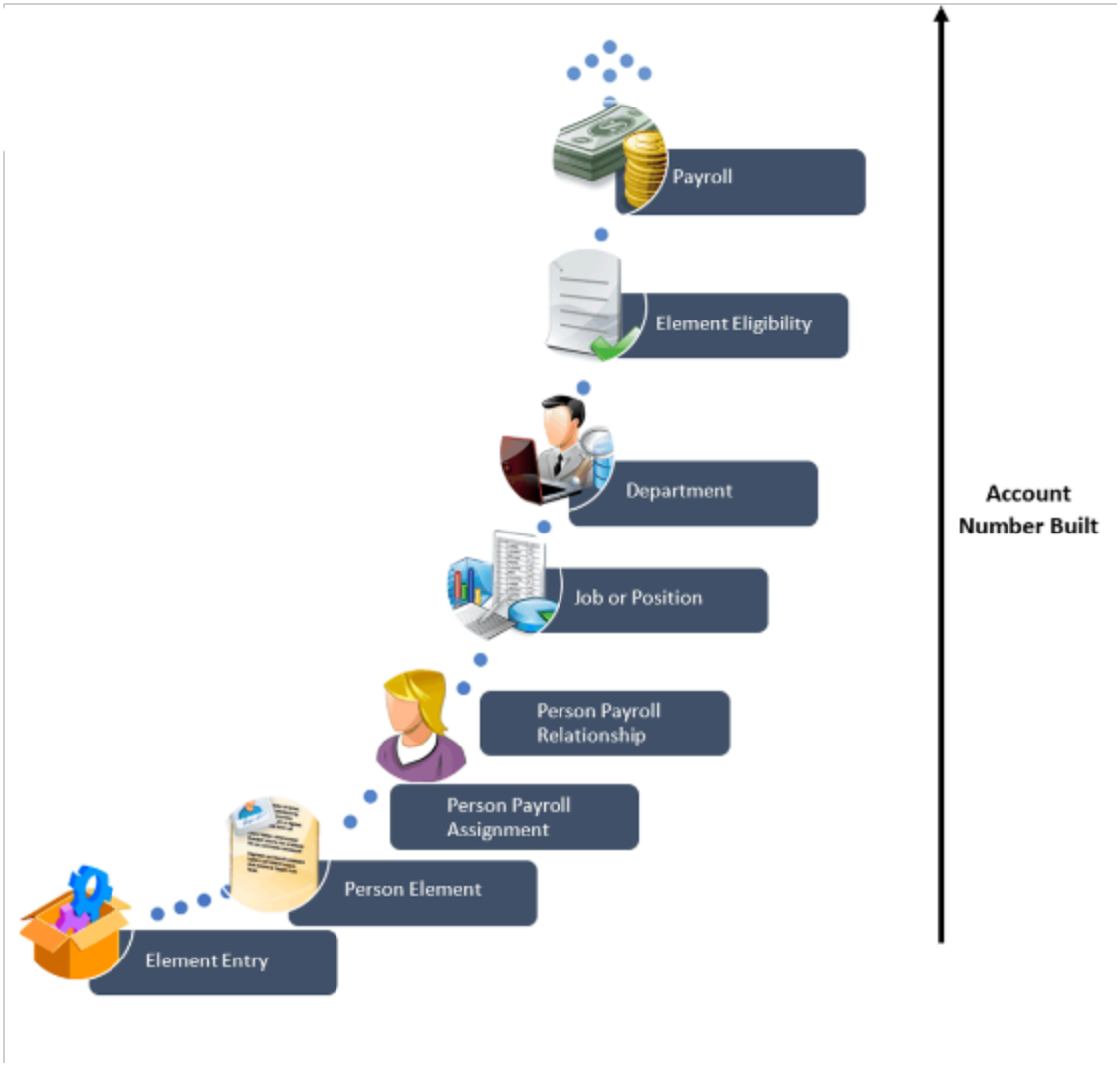
Set Up Payroll Costing

Overview of Loading Payroll Costing

Using HCM Data Loader, you can create payroll costing setup information for the different payroll accounts. For example, cost, offset, suspense, default, payroll liability, cash clearing, and cash accounts.

You can define costing at various levels, such as Payroll, Element Eligibility, Department, Job, Position, Person, and Payment Source.

This figure shows the different levels of the costing hierarchy.



You enter the account information and any overrides for the different levels of the cost hierarchy. The application builds the account number, starting with the lowest level (element entry) of the cost hierarchy and ending with the highest level (payroll). It checks each level sequentially until it finds a value. If it finds an invalid cost combination, it places the costing result in a suspense account.

Before You Begin

To load payroll costing records, do these tasks:

- Create Cost Allocation key flexfield. This flexfield creates a structure to capture the account codes used to create accounting entries and to track and report labor costs.
- Enable cost hierarchy with levels to support each cost allocation key flexfield cost account segment.
- Assign cost Allocation key flexfield to a legislative data group.

Payroll Costing Accounts

Load accounts to cost payroll run and payment results, and to store invalid and accounts that aren't allocated.

Using HCM Data Loader, you can do all of these tasks.

- Load cost accounts that store expenses and employer liabilities and charges.
- Create overrides by entering cost account numbers at lower levels of the cost hierarchy.
- Use priority accounts to cost elements that require the same account combination. For example, you use a priority account for an hourly earning element for laboratory work that's charged to a grant fund.
- Allocate a cost to more than one account by creating several accounts for an object. Specify the percentage to charge to each account. For example, allocate costs to split salary costs for a job shared between two cost centers.
- Load Offset accounts that create balancing entries required for double-entry bookkeeping.
- Set up suspense and default accounts at the payroll level and override them if required, with suspense and default accounts at the department level.

Related Topics

- Payroll Cost Allocation Key Flexfield Structure
- How Payroll Costing Components Integrate with Other Applications
- Payroll Setup for Costing Accounts

Example of Loading Element Eligibility Costing

Use HCM Data Loader to assign costing information at the element level. You cost the element's eligibility records when the results for the element affect net pay or employer liabilities.

When costing an element, you specify the values to cost, the type of costing to use, and then you offset the cost entries.

You can load the priority account that specifies the accounting at the element level. For example, your company funds the entire hourly earning element for work performed in a lab from a single account. In such a case, you can create a priority account and specify 100 for the percentage.

Let's suppose you cost a value for the same segment at higher levels in the costing hierarchy. In this case, though the value is defined at the element eligibility level, the application considers the higher segment value.

Example

The Payroll Manager of the Vision Corp organization wants to set up costing for the **Employer Union Pension Expense** element and distribute its costs to the earning elements of a distribution group.

Before You Begin

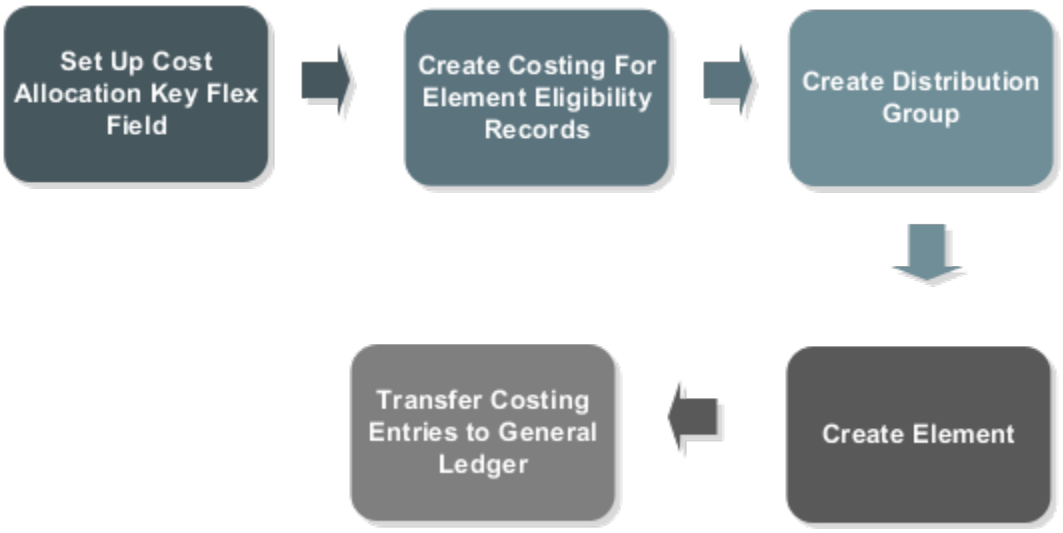
Before loading element costing, ensure that you do these steps:

1. Set up the Cost Allocation key flexfield.
2. Create costing for element eligibility records for each of the pensionable earnings elements, such as the regular wages and overtime wages.
3. Create the distribution group, **Pensionable Wages** that carries the costs of the distributed element. This group includes the employee's wage elements.

After setting up these prerequisites, you cost the employer portion of the pension liability by creating an **Employer Union Pension Expense** element.

For Distributed costing types, the calculation for distributed costing starts with the values loaded in the element eligibility costing record. The calculation derives values for other segments from the costing values on the associated entries of the distribution group. Further, it generates one costing result for each entry in the distribution group. Finally, you transfer the costing entries for the payroll run results to the General Ledger.

This figure shows how you cost details for an element.



Load these details for the **Employer Union Pension Expense** element:

Parameter	Value
Costing Type	Distributed
Transfer to General Ledger	Yes
Distribution Group	Pensionable Wages

Parameter	Value
Costed Input Value	Pay Value
Costed	Yes

When you're loading costing details for element eligibility, you must specify the costed input values.

```
METADATA|CostInfoV3|EffectiveEndDate|EffectiveStartDate|SourceType|ElementEligibilityName|ElementCode|
LinkInputName|LegislativeDataGroupName|CostableType|CostedFlag|TransferToGlFlag
MERGE|CostInfoV3|4712/12/31|2015/01/01|EL|Employer Union Pension Expense|Employer Union Pension Expense||
Vision_Corp_LDG|F||N
MERGE|CostInfoV3|4712/12/31|2015/01/01|LIV|Employer Union Pension Expense|Employer Union Pension Expense|Pay
 Value|Vision_Corp_LDG||Y|

METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|ElementEligibilityName|ElementCode|
LegislativeDataGroupName
MERGE|CostAllocationV3|4712/12/31|2015/01/01|EL|Employer Union Pension Expense|Employer Union Pension
 Expense|Vision_Corp_LDG
```

Load these cost account values for segments that you don't want the costing calculation to derive from distributed costing:

Parameter	Value
Fund (Segment3)	5220 Employer Union Pension Expense account
Appropriation (Segment 4)	1001
Funding Source (Segment 5)	1010

Load these values for segments in Offset Accounts. Here, the offset account is the payable liability account, and the balance sheet account numbers are for the Division and Department segments:

Parameter	Value
Division (Segment1)	00
Department (Segment2)	000
Natural Account (Segment6)	2152

```
METADATA|CostAllocationAccountV3|SourceType|ElementEligibilityName|ElementCode|LegislativeDataGroupName|
EffectiveDate|SourceSubType|Proportion|SubTypeSequence|ConcatenatedSegment|Segment3|Segment4|Segment5 MERGE|
CostAllocationAccountV3|EL|Employer Union Pension Expense|Employer Union Pension Expense|Vision_Corp_LDG
 |2015/01/01|BAL|1|1|00.000....2152||| MERGE|CostAllocationAccountV3|EL|Employer Union Pension Expense|
Employer Union Pension Expense|Vision_Corp_LDG |2015/01/01|COST|1|1||5220|1001|1010
```

Related Topics

- Overview of Loading Payroll Costing
- Element Costing Options
- How Distributed Costing Is Calculated
- How Payroll Costing Components Integrate with Other Applications

Example of Loading Costing Eligibility on Assignment Attributes

Use the HCM Data Loader to assign costing information at the costing eligibility on assignment attributes. Use this functionality to create the cost account information for payroll relationship elements using assignment attributes criteria, such as department, people group, or job.

Before loading the costing eligibility on assignment attributes for an element, it must have the costing defined on regular element with distributed costing type.

Example

Let's say, the Payroll Manager of the Vision Corp organization wants to set up costing for the Employer Union Pension Charges element, which is defined at payroll relationship level, to be costed to Natural Account 2601 for Payroll department and 2602 for Production department. Since the Department eligibility criteria isn't available for payroll relationship elements, you must use Costing Eligibility on Assignment Attributes functionality to create the costing information. Before you load the costing information on assignment attributes, you must first define or load the regular costing eligibility with no criteria (open link) and distributed costing type.

> **Note:** The cost account is only applicable for costing eligibility on assignment attributes. The offset account is used from the regular costing eligibility defined with distributed costing.

Use the Element Eligibility object with CostingLinkFlag set to Y to load the costing eligibility on assignment attributes.

Parameter	Value
Element Eligibility Name	Employer_Union_Pension_Charges_CostingEligib_PAYR
Department Code	UTILITIES_DEPT

Parameter	Value
Element Eligibility Name	Employer_Union_Pension_Charges_CostingEligib_PROD
Department Code	PRODUCTION_DEPT

```
METADATA|ElementEligibility|ElementId|ElementLinkId|EffectiveStartDate|EffectiveEndDate|
CalculationFormulaId|LegislativeDataGroupId|ElementEligibilityName|ValidationFormulaId|DefaultingFormulaId|
AutomaticEntryFlag|EmploymentCategory|BargainingUnitCode|LabourUnionMemberFlag|AllPayrollsEligibleFlag|
```

```
ElementCode|LegislativeDataGroupName|CollectiveAgreementIdCode|DepartmentCode|RelationshipTypeCode|
CalculationFormulaCode|ValidationFormulaCode|JobCode|DefaultingFormulaCode|GradeCode|PayrollStatUnitCode|
LegalEmployerCode|LocationCode|PayrollCode|PositionCode|JobSetCode|GradeSetCode|LocationSetCode|JobId|
GradeId|LocationId|PayrollId|PayrollStatUnitId|LegalEmployerId|PositionId|SourceSystemOwner|SourceSystemId|
GUID|RelationshipTypeId|CollectiveAgreementId|OrganizationId|PeopleGroup|CostingLinkFlag
MERGE|ElementEligibility|||2014/01/01|4712/12/31|||Employer_Union_Pension_Charges_CostingEligib_PAYR||
Employer Union Pension Charges|CRFL RRF LDG US1||UTILITIES_DEPT|Y
MERGE|ElementEligibility|||2014/01/01|4712/12/31|||Employer_Union_Pension_Charges_CostingEligib_PROD||
Employer Union Pension Charges|CRFL RRF LDG US1||PRODUCTION_DEPT||Y
```

Load these cost account values for segments for the element eligibility on costing attributes.

Parameter	Value
Element Eligibility	Employer_Union_Pension_Charges_CostingEligib_PAYR
Account (Segment 4)	2601

Parameter	Value
Element Eligibility	Employer_Union_Pension_Charges_CostingEligib_PROD
Account (Segment 4)	2602

Here's the sample dat file to load Cost Allocation.

```
METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|ElementEligibilityName|
DepartmentName|JobCode|PositionCode|SetCode|PayrollCode|PaymentSourceCode|Name|AssignmentNumber|
PayrollRelationshipNumber|TermNumber|ElementCode|MultipleEntryCount|LegislativeDataGroupName|
GUID|SourceSystemId|SourceSystemOwner|AssignmentId(SourceSystemId)|PayrollId(SourceSystemId)|
PaymentMethodId(SourceSystemId)|ElementEntryId(SourceSystemId)|ElementLinkId(SourceSystemId)|
OrganizationId(SourceSystemId)|JobId(SourceSystemId)|PositionId(SourceSystemId)|
OrganizationPaymentMethodCode|BusinessUnitName|EntryType
MERGE|CostAllocationV3|4712/12/31|2014/01/01|EL|Employer_Union_Pension_Charges_CostingEligib_PAYR||Employer
 Union Pension Charges||CRFL RRF LDG US1||
MERGE|CostAllocationV3|4712/12/31|2014/01/01|EL|Employer_Union_Pension_Charges_CostingEligib_PROD||Employer
 Union Pension Charges||CRFL RRF LDG US1||
```

Here's the sample dat file to load Cost Allocation Account.

```
METADATA|CostAllocationAccountV3|SourceType|ElementEligibilityName|DepartmentName|JobCode|PositionCode|
SetCode|PaymentSourceCode|PayrollCode|Name|AssignmentNumber|PayrollRelationshipNumber|TermNumber|
ElementCode|MultipleEntryCount|LegislativeDataGroupName|EffectiveDate|SourceSubType|Proportion|
SubTypeSequence|ConcatenatedSegment|Segment1|Segment2|Segment3|Segment4|Segment5|GUID|SourceSystemId|
SourceSystemOwner|AssignmentId(SourceSystemId)|PayrollId(SourceSystemId)|PaymentMethodId(SourceSystemId)|
ElementEntryId(SourceSystemId)|ElementLinkId(SourceSystemId)|OrganizationId(SourceSystemId)|
JobId(SourceSystemId)|PositionId(SourceSystemId)|OrganizationPaymentMethodCode|BusinessUnitName|EntryType
MERGE|CostAllocationAccountV3|EL|Employer_Union_Pension_Charges_CostingEligib_PAYR||Employer Union Pension
 Charges||CRFL RRF LDG US1|2014/01/31|COST|1|1|||2601||
MERGE|CostAllocationAccountV3|EL|Employer_Union_Pension_Charges_CostingEligib_PROD||Employer Union Pension
 Charges||CRFL RRF LDG US1|2014/01/31|COST|1|1|||2602||
```

Example of Loading Costing of Payroll

Use HCM Data Loader to assign account information to payrolls. You can select this highest level for segments that seldom change for the people assigned to the payroll, such as company and line of business.

As a best practice, set up a suspense and default account at the payroll level. Otherwise, you must set up a suspense and default account for every department to ensure you charge invalid costs to an account.

Example

Vision Corp has multiple payrolls, such as weekly and monthly and loads costing details at the payroll level. And they use the costing information to report labor cost and generate journal entries for all their employees for the specified payroll.

Load these costing details for the monthly payroll Vision Corp Monthly.

Parameter	Value
Fund (Segment1)	All Funds; 0001
Appropriation (Segment2)	Salaries Appropriation;1001
Funding Source(Segment3)	Original Funding Source;1010
Department(Segment4)	General Government Agencies;1000
Organization(Segment5)	Payroll; 1003
Program(Segment6)	Balance Sheet; 0000
Object (Segment7)	Unspecified;0000
Project(Segment8)	Unspecified; 0000

Load values for both suspense and default accounts.

Parameter	Value
Fund(Segment1)	Unspecified; 0000
Appropriation(Segment2)	Balance Sheet; 0000
Funding Source(Segment3)	Balance Sheet; 0000

Parameter	Value
Department(Segment4)	Balance Sheet; 0000
Organization (Segment5)	Balance Sheet; 0000
Program(Segment6)	Balance Sheet; 0000
Object (Segment7)	Unspecified; 0000
Project(Segment8)	Unspecified; 0000

This example .dat file loads the costing allocation details for the monthly payroll.

```
METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|PayrollCode|
LegislativeDataGroupName
MERGE|CostAllocationV3|4712/12/31|2010/01/01|PAY|Vision Corp Monthly| Vision Corp
```

This example .dat file loads the costing account details for the monthly payroll.

```
METADATA|CostAllocationAccountV3|SourceType|PayrollCode|LegislativeDataGroupName|EffectiveDate|
SourceSubType|Proportion|SubTypeSequence|ConcatenatedSegment|Segment1|Segment2|Segment3|Segment4|Segment5|
Segment6|Segment7|Segment8
MERGE|CostAllocationAccountV3|PAY|Vision Corp Monthly|Vision Corp|2010/01/01|COST|1|1||0001|1001|1010|1000|
1003|0000|0000|0000
MERGE|CostAllocationAccountV3|PAY|Vision Corp Monthly|Vision Corp|2010/01/01|SUSP|1|1|
0000.0000.0000.0000.0000.0000.0000.0000||
MERGE|CostAllocationAccountV3|PAY|Vision Corp Monthly|Vision Corp|2010/01/01|DFLT|1|1|
0000.0000.0000.0000.0000.0000.0000.0000||
```

Related Topics

- How Payroll Costing Components Integrate with Other Applications
- Payroll Setup for Costing Accounts
- Payroll Cost Allocation Key Flexfield Structure

Example of Loading Costing of Departments

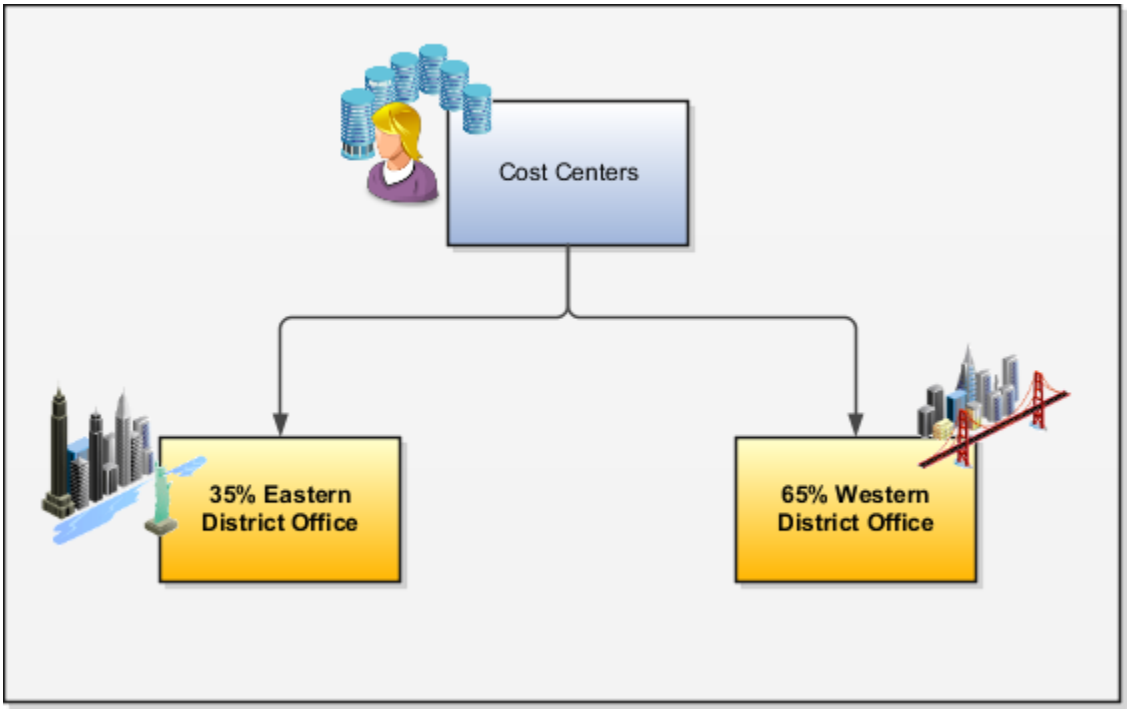
Use HCM Data Loader to define payroll costing rules at the Department level. For example, in a large enterprise you might set up default accounts for departments where the managers commonly review and resolve their department's expenses.

When you load costs at the department level, you can specify the percentage each account receives of the cost. If the total allocation isn't 100 percent, then the application places the invalid cost in a default account.

Example

Vision Corporation's Payroll Manager wants to allocate costing at the department level to two cost centers.

As this figure shows, she allocates 35 percent of the costs of the Administration department to the Eastern District Office. She allocates the remainder to the Western District Office.



The Payroll Manager loads these costing details for the Administration department to two district officers as below:

Parameter	Eastern District Office	Western District Office
Percentage	35	65
Fund (Segment1)	Pooled Cash Fund: 0900	Pooled Cash Fund: 0901
Appropriation (Segment2)	1001	1001
Funding Source (Segment3)	1010	1011

Further, she uses the payroll suspense account to store costed payroll run results and prepayment results with invalid account combinations. She loads these suspense account details.

Parameter	Value
Fund (Segment1)	All Funds; 0001
Appropriation (Segment2)	Balance Sheet; 0000
Funding Source (Segment3)	Balance Sheet; 0000
Department (Segment4)	Balance Sheet; 0000

Parameter	Value
Organization (Segment5)	Balance Sheet; 0000
Object (Segment7)	Unspecified; 0000
Project (Segment8)	Unspecified; 0000

As Vision Corporation's department managers review and resolve their department expenses, the Payroll Manager sets up default accounts for the Administration department. She loads these values for the default account.

Parameter	Value
Fund (Segment1)	Unspecified; 0000
Appropriation (Segment2)	Balance Sheet; 0000
Funding Source (Segment3)	Balance Sheet; 0000
Department (Segment4)	Balance Sheet; 0000
Organization (Segment5)	Balance Sheet; 0000
Program (Segment6)	Balance Sheet; 0000
Object (Segment7)	Unspecified; 0000
Project (Segment8)	Unspecified; 0000

This example dat file loads the costing details for the two departments.

```
METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|DepartmentName|
LegislativeDataGroupName
MERGE|CostAllocationV3|4712/12/31|2010/01/01|ORG|Administration|Vision Corp

METADATA|CostAllocationAccountV3|SourceType|DepartmentName|LegislativeDataGroupName|EffectiveDate|
SourceSubType|Proportion|SubTypeSequence|ConcatenatedSegment|Segment1|Segment2|Segment3|Segment4|Segment5|
Segment6|Segment7|Segment8
MERGE|CostAllocationAccountV3|ORG|Administration|Vision Corp|2010/01/01|COST|0.35|1||0900|1001|1010||
MERGE|CostAllocationAccountV3|ORG|Administration|Vision Corp|2010/01/01|COST|0.65|2||0901|1333|220||
MERGE|CostAllocationAccountV3|ORG|Administration|Vision Corp|2010/01/01|SUSP|1|1|
0001.0000.0000.0000.0000.0000.0000.0000||
MERGE|CostAllocationAccountV3|ORG|Administration|Vision Corp|2010/01/01|DFLT|1|1|
0000.0000.0000.0000.0000.0000.0000.0000||
```

Related Topics

- Allocate Costs to Accounts
- Payroll Setup for Costing Accounts

Example of Loading Costing of Jobs

Use HCM Data Loader to define payroll costs based upon the job category, such as cost centers or project funds. Let's assume that you want to compare cost results for jobs in different cost centers.

Then, you allocate costs at the job level to generate the cost results based on the job.

You can load cost account segments to override those segments defined at other levels. If you divide costs among accounts, you can specify the percentage each account receives of the cost. If the total allocation isn't 100 percent, then the application places the costs that aren't allocated to a default account.

Example

Assume that you want to cost accounts to store expenses and employer liabilities and charges for the job Senior Manager.

This table shows the job details that you want to cost for the employee.

Parameter	Value
Percentage	100
Fund (Segment1)	Pooled Cash Fund: 0900
Appropriation (Segment2)	1001

Use these dat files to load the cost segments for the job Senior Manager:

```
METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|JobCode|SetCode|
LegislativeDataGroupName
MERGE|CostAllocationV3|4712/12/31|2010/01/01|JOB|Senior Manager|COMMON|Vision Corp

METADATA|CostAllocationAccountV3|SourceType|JobCode|SetCode|LegislativeDataGroupName|EffectiveDate|
SourceSubType|Proportion|SubTypeSequence|Segment1|Segment2
MERGE|CostAllocationAccountV3|JOB|Senior Manager|COMMON|Vision Corp|2010/0101|COST|1|0900|1001
```

Related Topics

- Payroll Setup for Costing Accounts
- Allocate Costs to Accounts
- Allocate Costs to Several Accounts

Example of Loading Costing of Positions

If you're using position management at your enterprise to track the cost of turnover to the enterprise, then you allocate costs at the position level.

For example, you want to compare cost results for positions in different cost centers. So, you allocate the costs at the position level to generate cost results based on the position.

Let's consider this example. Vision Corp uses position management and wants to cost these account details for the position Senior Product Manager in Utilities business unit.

Parameter	Value
Percentage	100
Fund (Segment1)	Government Fund;1000
Appropriation (Segment2)	Balance Sheet; 0000
Funding Source (Segment3)	Funding Source; 2000

Use this dat file to cost the position details.

```
METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|PositionCode|
LegislativeDataGroupName|BusinessUnitName
MERGE|CostAllocationV3|4712/12/31|2010/01/01|POS|Senior Product Manager|Vision Corp|Utilities

METADATA|CostAllocationAccountV3|SourceType|PositionCode|LegislativeDataGroupName|EffectiveDate|
SourceSubType|Proportion|SubTypeSequence|Segment1|Segment2|Segment3|BusinessUnitName
MERGE|CostAllocationAccountV3|POS|Senior Product Manager|Vision Corp|2010/01/01|COST|1|1|100|0000|2000|
Utilities
```

Related Topics

- Payroll Setup for Costing Accounts
- Allocate Costs to Accounts
- Allocate Costs to Several Accounts

Example of Loading Payment Source Costing

Use HCM Data Loader to enter account information for your payment sources.

Imagine a scenario where a delay exists between the date the payment is issued by using a cheque and the date the cheque is cleared. In such a case, your company wants to create a cash clearing account to track payments.

Using HCM Data Loader, you load a payment source and create liability, cash clearing, and cash accounts for it.

Note: You load the same account information that you use for the cash and cash clearing account that you created in the General Ledger.

This table summarizes the values that you load for the various accounts:

Parameter	Cash Account	Cash Clearing Account	Liability Account
Fund (Segment1)	All Funds; 0001	Pooled Cash Fund;0900	All Funds; 0001
Appropriation (Segment2)	Salaries Appropriation ; 1001	Salaries Appropriation; 1002	Salaries Appropriation; 1002
Funding Source (Segment3)	Original Funding Source; 1010	Funding Source; 1999	Balance Sheet; 0000
Department (Segment4)	Progress Top Level; 0001	General Government Agencies; 1000	Balance Sheet; 0000
Organization (Segment5)	Administration; 1001	Administration; 1001	Payroll; 1003
Program (Segment6)	Programs; 0001	Programs; 0001	Legislative;1100
Object (Segment7)	Cash; 11110	Object; 11140	On Account Cash; 11150
Project (Segment8)	Administrative Projects; 100000	Capital Project; 100001	Broadway Extension; 100004

This dat file loads these account values for the payment source Vision Corp Check PS in Vision Corp Check PM payment method:

```
METADATA|CostInfoV3|EffectiveEndDate|EffectiveStartDate|SourceType|PaymentSourceCode|
OrganizationPaymentMethodCode|LegislativeDataGroupName|CostableType|CostedFlag|TransferToGlFlag|GUID|
SourceSystemId|SourceSystemOwner
MERGE|CostInfoV3|4712/12/31|2010/01/01|PM|Vision Corp Check PS|Vision Corp Check PM|Vision Corp|Y|Y|Y|

METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|PaymentSourceCode|
OrganizationPaymentMethodCode|LegislativeDataGroupName|GUID|SourceSystemId|SourceSystemOwner
MERGE|CostAllocationV3|4712/12/31|2010/01/01|PM|Vision Corp Check PS|Vision Corp Check PM|Vision Corp||

METADATA|CostAllocationAccountV3|SourceType|PaymentSourceCode|OrganizationPaymentMethodCode|
LegislativeDataGroupName|EffectiveDate|SourceSubType|Proportion|SubTypeSequence|ConcatenatedSegment| GUID|
SourceSystemId|SourceSystemOwner
MERGE|CostAllocationAccountV3|PM|Vision Corp Check PS|Vision Corp Check PM|Vision Corp|2010/01/01|CLRNG|1|1|
0900.1002. 1999.1000.1001.0001.11140.100001||
MERGE|CostAllocationAccountV3|PM|Vision Corp Check PS|Vision Corp Check PM|Vision Corp|2010/01/01|CASH|1|1|
0001.1001.1010.0001.1001.0001.11110.100000||
MERGE|CostAllocationAccountV3|PM|Vision Corp Check PS|Vision Corp Check PM|Vision Corp|2010/01/01|LBLTY|1|1|
0001.1002.0000.0000.1003.1100.11150.100004||
```

Related Topics

- Overview of Loading Payroll Costing
- Payroll Setup for Costing Accounts
- How Payroll Cost Results are Calculated
- Payroll Setup Tasks to Transfer Costs to General Ledger

Example of Loading Person Costing

Use HCM Data Loader to manage costing at the person level to track costs for people in your enterprise.

You can either cost all the elements that the person is eligible to receive or cost individual elements. You allocate costs to single or multiple accounts.

Let's suppose you start a new project and want to track the costs incurred by the employees reassigned temporarily to the project. To monitor these costs, you can set up costing at the person level for these employees.

When you load costs at the person level, you can specify the percentage each account receives of the cost. If the total allocation isn't 100 percent, then the application places the cost that's not allocated in a default account.

Using HCM Data Loader, you can load costing for a person at the assignment and payroll relationship levels. Further, you can allocate costing for an element at the assignment or payroll relationship level for a person.

Example of Loading Person Costing at Payroll Relationship Level

Let's consider this example where you cost Vijay's elements to the Special Revenue Fund. When you define the costs at the payroll relationship level, you need not define the costs at each element level for these segments.

Load these details to allocate costs for Vijay (Payroll Relationship Number: 12345678) at the payroll relationship level.

Parameter	Value
Percentage	100
Fund (Segment1)	Special Revenue Fund (1100)
Appropriation (Segment2)	Salary Appropriation (1002)
Funding Source (Segment3)	2000
Department (Segment4)	2000
Organization (Segment5)	Payroll (1003)

This example dat file loads the costing details for the person at the payroll relationship level.

```
METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|PayrollRelationshipNumber|
LegislativeDataGroupName
MERGE|CostAllocationV3|4712/12/31|2010/01/01|PREL|12345678|Vision Corp
METADATA|CostAllocationAccountV3|SourceType|PayrollRelationshipNumber|LegislativeDataGroupName|
EffectiveDate|SourceSubType|Proportion|SubTypeSequence|Segment1|Segment2|Segment3|Segment4|Segment5
MERGE|CostAllocationAccountV3|PREL|12345678|Vision Corp|2010/01/01|COST|1|1|1100|1002|2000|2000|1003
```

Related Topics

- Example of Loading Person Costing at Assignment Level
- Example of Loading Person Element Costing at Assignment Level
- Example of Loading Person Element Costing at Payroll Relationship Level

Example of Loading Person Costing at Assignment Level

As Vision Corp encourages employees to work across multiple projects, employee's costs should be allocated in the same way.

In this example, you allocate costing at the assignment level for a person who divides the time worked between two managers at different cost centers. Further, you override costing for a specific element at the assignment level.

Lynda Jones writes advertisements for the Advertising division. For the next few months, she will spend 40 percent of his time in designing customer preference surveys for the Market Research division. Now, you split Lynda's cost between Advertising and Market Research divisions, which belong to different cost centers.

Assign these percentages to the two departments:

Percent	Department
40	3180 (Market Research)
60	3190 (Advertising)

The assignment number of Lynda is 12345.

This example .dat file loads the costing details for a person at the assignment level:

```
METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|AssignmentNumber|
LegislativeDataGroupName
MERGE|CostAllocationV3|4712/12/31|2010/01/01|ASG|12345|Vision Corp

METADATA|CostAllocationAccountV3|SourceType|AssignmentNumber|LegislativeDataGroupName|EffectiveDate|
SourceSubType|Proportion|SubTypeSequence|Segment4
MERGE|CostAllocationAccountV3|ASG|12345|Vision Corp|2010/01/01|COST|0.4|1|3180
MERGE|CostAllocationAccountV3|ASG|12345|Vision Corp|2010/01/01|COST|0.6|2|3190
```

Related Topics

- Example of Loading Person Costing at Payroll Relationship Level
- Example of Loading Person Element Costing at Assignment Level
- Example of Loading Person Element Costing at Payroll Relationship Level

Example of Loading Person Element Costing at Assignment Level

In this example, all elements of the employee John go to the cost center Payroll and are defined at the payroll relationship level.

However, in a particular period, the Vision Corp pays a bonus to John and the costs of the bonus should go to his home department, which is Finance.

Here, you define the Finance cost center for the Bonus element using the assignment costing element.

Parameter	Value
Element Name	Vision_Corp_Bonus
Percentage	100
Funding Source	1010
Department	3120 Finance Department

The employee's assignment number is 12345 and the Payroll Relationship number is 12345678.

This .dat file loads the costing details for the two departments.

```
METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|AssignmentNumber|
PayrollRelationshipNumber|ElementCode|LegislativeDataGroupName
MERGE|CostAllocationV3|4712/12/31|2010/01/01|AET|12345|12345678|Vision_Corp_Bonus|Vision Corp

METADATA|CostAllocationAccountV3|SourceType|AssignmentNumber|ElementCode|PayrollRelationshipNumber|
LegislativeDataGroupName|EffectiveDate|SourceSubType|Proportion|SubTypeSequence|Segment3|Segment4
MERGE|CostAllocationAccountV3|AET|12345|Vision_Corp_Bonus|12345678|Vision Corp|2010/01/01|COST|1|1|1010|3120
```

Related Topics

- Example of Loading Person Costing at Payroll Relationship Level
- Example of Loading Person Costing at Assignment Level
- Example of Loading Person Element Costing at Payroll Relationship Level

Example of Loading Person Element Costing at Payroll Relationship Level

In this example, you update the funding source for the ANC_ELE_COMP_VISION Entitlement element of Mandy Steward as University Grants for research.

The application derives the funding source and department for all other elements from element eligibility costing and department level costing. Then, you can define these segments at payroll relationship element level.

For the ANC_ELE_COMP_VISION Entitlement element, load these costs at the payroll relationship level for Mandy.

Parameter	Value
Percentage	100
Funding Source	University Grants for research 2003
Department	Education 2110
Organization	Payroll (1003)

The employee's payroll relationship number is 12345678.

This example .dat file loads the costing details at the payroll relationship level for the person.

```
METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|PayrollRelationshipNumber|
ElementCode|LegislativeDataGroupName
MERGE|CostAllocationV3|4712/12/31|2010/01/01|PRET|12345678|ANC_ELE_COMP_VISION Entitlement element|Vision
 Corp

METADATA|CostAllocationAccountV3|SourceType|PayrollRelationshipNumber|ElementCode|LegislativeDataGroupName|
EffectiveDate|SourceSubType|Proportion|SubTypeSequence|Segment3|Segment4|Segment5
MERGE|CostAllocationAccountV3|PRET|12345678|ANC_ELE_COMP_VISION Entitlement element|Vision Corp|2010/01/01|
COST|1|1|2003|2110|1003
```

Related Topics

- Example of Loading Person Costing at Payroll Relationship Level
- Example of Loading Person Costing at Assignment Level
- Example of Loading Person Element Costing at Assignment Level

Example of Loading Element Entry Costing

David works for Vision Corp and all of his elements go to his home department Payroll (3001).

However, last Monday, David has provided services to HR department (3002) for 8 hours. Now, these hours should be costed to the HR department. The services are categorized as consulting (4001).

Using HCM Data Loader, load these element entry costing values.

Parameter	Value
Effective Start Date	01-Jan-2011
Payroll Relationship Number	955160008175766
Person Number	955160008175766
Element	Hourly Salary
Department (Segment4)	3002
Program (Segment6)	4001

The employee's assignment number is 12345 and the payroll relationship number is 12345678.

This example .dat file loads the element entry level costing details for David.

```
METADATA|CostAllocationV3|EffectiveEndDate|EffectiveStartDate|SourceType|AssignmentNumber|ElementCode|
MultipleEntryCount|LegislativeDataGroupName|EntryType
MERGE|CostAllocationV3|4712/12/31|2010/01/01|EE|12345|Hourly Salary|1|Vision Corp|E

METADATA|CostAllocationAccountV3|SourceType|AssignmentNumber|ElementCode|MultipleEntryCount|
LegislativeDataGroupName|EffectiveDate|SourceSubType|Proportion|SubTypeSequence|Segment4|Segment6|EntryType
MERGE|CostAllocationAccountV3|EE|12345|Hourly Salary|1|Vision Corp|2010/01/01|COST|1|1|3002|4001|E
```

Related Topics

- Payroll Cost Allocation Key Flexfield Structure
- How Payroll Costing Components Integrate with Other Applications
- Payroll Setup for Costing Accounts

Load Costing Validations

Overview of Loading Costing Cross-Validation Rules

Using HCM Data Loader, you can create costing cross-validation rules that define validation across segments. These rules enforce whether a value of a particular segment can be combined with specific values of other segments to form a new combination.

Define cross-validation rules for payroll costing key flexfield segments. After enabling the rules, use them to determine the valid cost account combinations that you can create during the costing process.

Example: Use validation rules to check if you can combine a particular segment value with values in other segments to form a new cost account combination. Also, you can use these rules to prevent any invalid combinations that go to General Ledger.

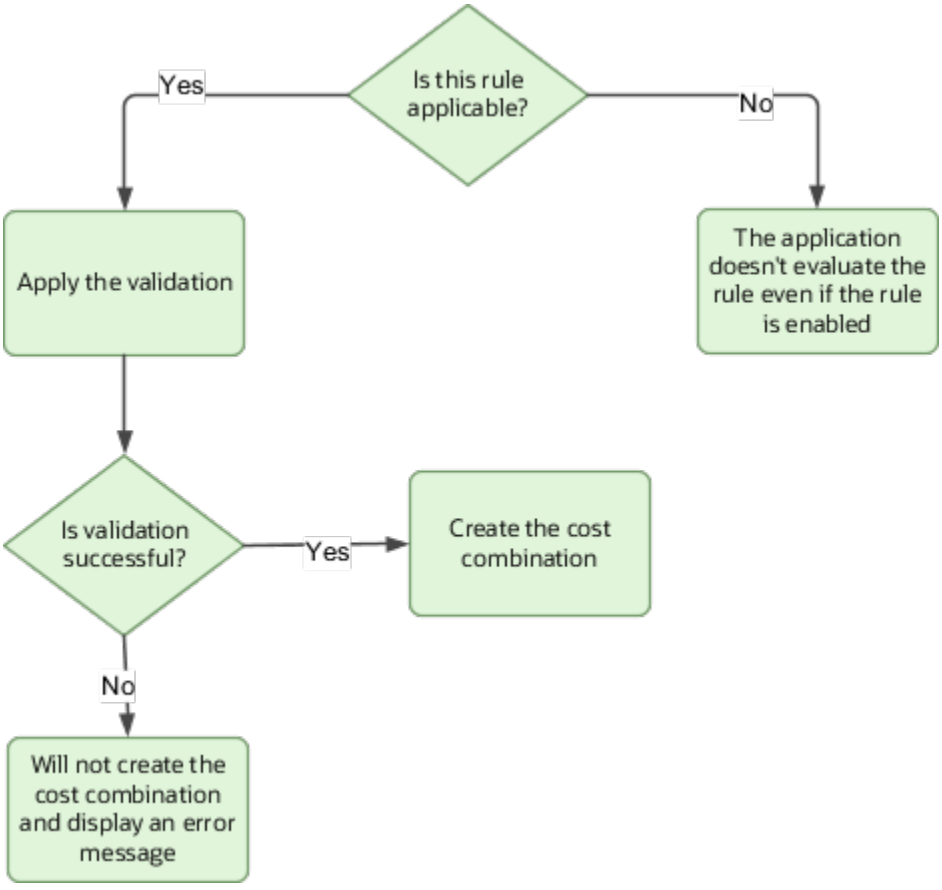
Components of a Cross Validation Rule

A cross-validation rule includes a condition filter and a validation filter. The application evaluates the rule using this logical order:

If the condition filter is satisfied, then apply the validation filter.

The condition filter describes the event when the application evaluates the rule. As this figure shows, depending upon the event, the application applies the validation filter and creates the combination.

- If the event specified in the condition filter isn't applicable, then the application doesn't evaluate the rule even if it's enabled.
- If the event specified in the condition filter is applicable, the validation filter condition must be applied before the application can create a combination.



Example

Your organization has determined that a certain company value called Operations can't use a specific cost center called Marketing.

As this figure shows, you can define a cross-validation rule to validate your combinations.

1. The rule evaluates the company Condition Filter.
2. When company is equal to Operations, the rule evaluates the cost center validation filter.
3. When cost center is equal to Marketing, the rule prevents a combination from being created.

Note: Such a rule doesn't affect the creation of combinations with Marketing cost center and company values other than Operations.

This table lists the various segment codes that you can use in your DAT files.

Segment Name	Segment Code
Company	COMPANY
Business Unit	Business Unit
Cost Center	COST_CENTER
Activity Code	ACTIVITY_CODE
Activity Code Charged	ACTIVITY_CODE_CHARGED
Location	LOCATION

Note: Use Segment Code in the DAT file while loading the data. For example, you should use "COST_CENTER" in the DAT file instead of "Cost Center". If the segment code has spaces in it, use it with double quotes. For example, you should use "Business Unit" while defining a rule using business unit segment.

Related Topics

- Example of Loading Cross-Validation Rules
- Examples of Loading Cross-Validation Rules With Operators
- Examples of Updating Costing Cross-Validation Rules
- Example of Updating Cost Code Combinations

Example of Loading Cross-Validation Rules

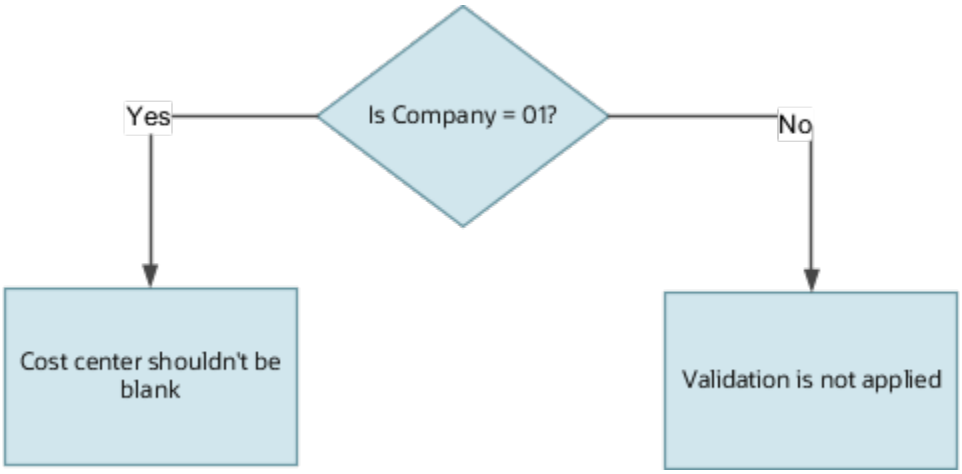
Use HCM Data Loader to create costing cross-validation rules that define validation across segments. Recollect that a cross-validation rule includes a condition filter and a validation filter. The application first validates the condition filter and if it's true, then applies the validation filter.

Let's consider these examples where we discuss about how you can use HCM Data Loader to load cross-validation rules.

Example 1: Cost Center is Required

Your organization has a cross-validation rule that prevents you from loading blank Cost-Center value for 01 Company. As the first step, the application evaluates the condition filter that the company equals 01. If the condition filter is satisfied, then the application applies the validation filter that the cost center should not be blank.

This illustration shows how the application evaluates the condition filter and later applies the validation filter accordingly for the cross-validation rule.



Use this CostKffValidationRule dat file to apply these condition and validation filters:

```
METADATA|CostKffValidationRule|StructureInstanceCode|RuleCode|RuleDescription|ErrorMessageText|EnabledFlag|
StartDateActive|EndDateActive|ConditionFilterText|ValidationFilterText
MERGE|CostKffValidationRule|HDL_COST_CVR_COST_KFF_INST|<RuleCode>|<ruleDesc>|<errorMsgTxt>|Y|2010/01/01|
4712/12/31|COMPANY = 01|COST_CENTER ISNOTBLANK
```

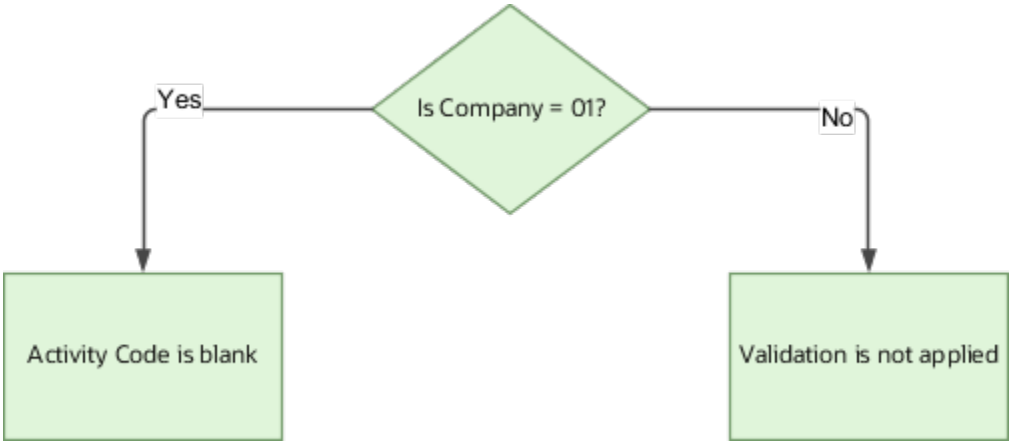
Note: For each new rule in the dat file, you must supply the RuleCode, RuleDesc and ErrorMsgTxt. For example, in the above scenario, the dat file looks as below:

```
MERGE|CostKffValidationRule|HDL_COST_CVR_COST_KFF_INST|CVR_COMP01COSTBLANKVAL|Company and Cost Center
 Validation|Cost Center must entered for Company 01Y|2010/01/01|4712/12/31|COMPANY = 01|COST_CENTER
 ISNOTBLANK
```

Example 2: Activity Code is Prohibited

Vision Corp prohibits activity code for 01 Company. The application evaluates the condition filter that the company equals 01. If the condition filter is satisfied, then the application applies the validation filter that the activity code should be blank for that company code.

This illustration shows how the application evaluates the condition filter and later applies the validation filter accordingly for the cross-validation rule.



Use this dat file to apply the condition and validation filters:

```
METADATA|CostKffValidationRule|CrossValidationRuleId|StructureInstanceCode|RuleCode|RuleDescription|
ErrorMessageText|EnabledFlag|StartDateActive|EndDateActive|ConditionFilterText|ValidationFilterText
MERGE|CostKffValidationRule|22|HDL_COST_CVR_COST_KFF_INST|<RuleCode>|<ruleDesc>|<errorMsgTxt>|Y|2010/01/01|
4712/12/31|Company=01|ACTIVITY_CODE ISBLANK
```

Example 3: Activity Code First Position When Activity Code Isn't Null

For Company Code = 01, if the activity code isn't null, then its first position must be one of these values: 0,1,2,3 or 4.

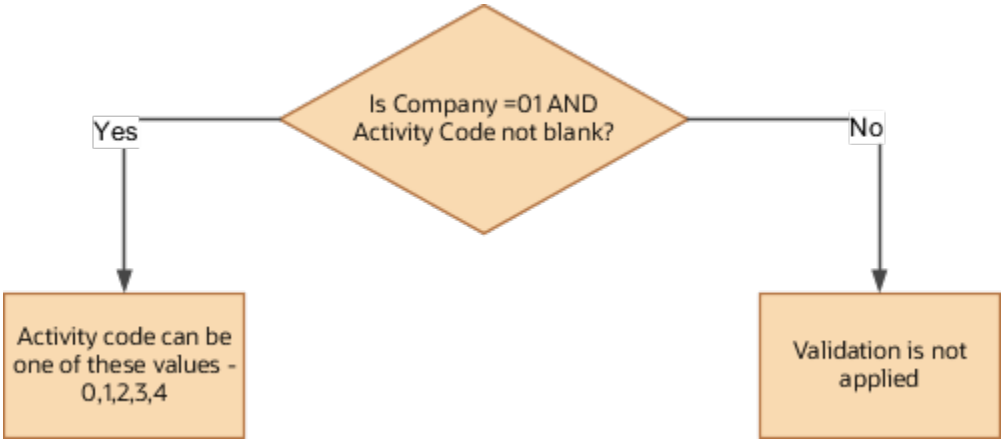
The application evaluates this condition:

(Company Equals 01) AND (Activity Code IS NOT BLANK)

If the condition filter is satisfied, then the application applies this validation filter:

(Activity Code STARTSWITH 0) or (Activity Code STARTSWITH 1) or (Activity Code STARTSWITH 2) or (Activity Code STARTSWITH 3) or (Activity Code STARTSWITH 4)

This illustration shows how the application evaluates the condition filter and later applies the validation filter accordingly for the cross-validation rule.



Use this dat file to apply the condition and validation filters:

```
METADATA|CostKffValidationRule|CrossValidationRuleId|StructureInstanceCode|RuleCode|RuleDescription|
ErrorMessageText|EnabledFlag|StartDateActive|EndDateActive|ConditionFilterText|ValidationFilterText
```

```
MERGE|CostKffValidationRule|23|HDL_COST_CVR_COST_KFF_INST|<RuleCode>|<ruleDesc>|<errorMsgTxt>|N|2010/01/01|
4712/12/31|Company = 01,ACTIVITY_CODE ISNOTBLANK|ACTIVITY_CODE STARTSWITH 0,ACTIVITY_CODE STARTSWITH
 1,ACTIVITY_CODE STARTSWITH 2,ACTIVITY_CODE STARTSWITH 3,ACTIVITY_CODE STARTSWITH 4
```

Related Topics

- Overview of Loading Costing Cross-Validation Rules
- Examples of Loading Cross-Validation Rules With Operators
- Examples of Updating Costing Cross-Validation Rules

Examples of Loading Cross-Validation Rules With Operators

When creating cross-validation rules, the application uses OR operator if the same segment is used with different values in Condition or Validation filters. Further, the application uses AND operators for different segments in condition or validation filters.

Let's consider these examples with OR and AND operators.

If you define a condition filter for the company segment values, Operations and Production, then the application evaluates the rule as shown in this statement:

Company = Operations (OR) Company = Production

If you define a condition filter for Company = Operation and Business Unit = Albany, then the application evaluates the rule like this:

Company = Operations (AND) Business Unit = Albany

The supported operators are shown in this table:

Operator	Description
STARTSWITH	Starts with
ENDSWITH	Ends with
=	Equals
<>	Does not equal
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
BETWEEN (Using '-' to separate two values)	Between

Operator	Description
NOTBETWEEN	Not Between
CONTAINS	Contains
DOESNOTCONTAIN	Does not contain
ISBLANK	Is blank
ISNOTBLANK	Is not blank

Example 1: Location Depends Upon Activity

If the Activity Code is equal to 2200, then Location can't be either 2201 or 22EE.

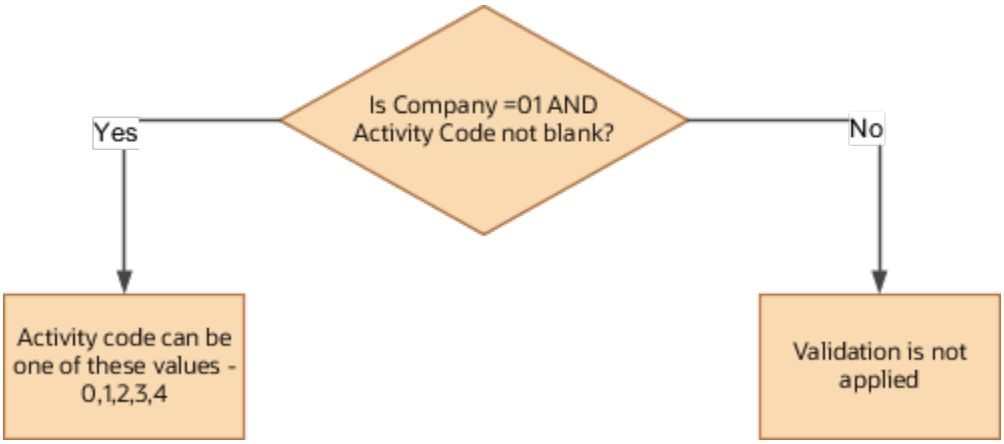
The application evaluates this condition:

Activity Code = 2200

If the condition filter is satisfied, then the application applies this validation filter:

(Location NOTEQUAL 2201) or (Location NOTEQUAL 22EE)

This illustration shows how the application evaluates the condition filter and later applies the validation filter accordingly for the cross-validation rule.



Use this CostKffValidationRule.dat file to apply the condition and validation filters:

```
METADATA|CostKffValidationRule|StructureInstanceCode|RuleCode|RuleDescription|ErrorMessageText|EnabledFlag|
StartDateActive|EndDateActive|ConditionFilterText|ValidationFilterText
MERGE|CostKffValidationRule|31|HDL_COST_CVR_COST_KFF_INST|ACTIVITY_LOCATION_RULE|Rule to Check Activity and
 Location|Activity and Rule not Satisfy Rule Check|Y|2010/01/01|4712/12/31|ACTIVITY_CODE = 2200|LOCATION <>
 2201,LOCATION <> 22EE
```

In the above example, if the cost combination has Activity Code = 2200, then the process checks whether the Location is 2201 or 22EE. If the Location is one of these values, then the cost combination doesn't get created as it isn't a valid combination. Any value other than 2201 or 22EE for Location, is considered as valid for Activity Code = 2200 and the cost code combination is created.

Example 2: Activity Code Validation Based On Activity Code and Activity Code Charged

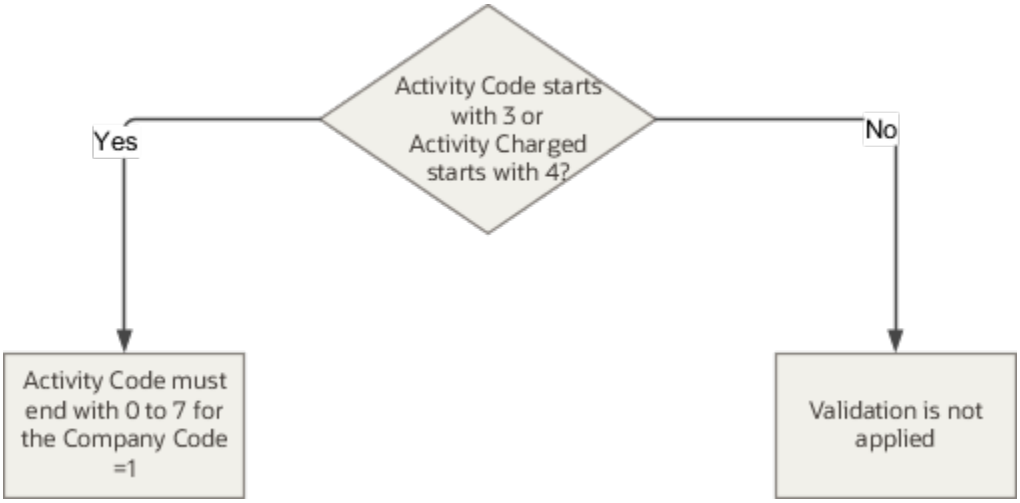
Let's assume that the Activity Code starts with 3 or the Activity Code Charged starts with 4. Then, the Activity Code must end with {0-7} for the company code =01.

Let's assume that the Activity Code starts with 3 or the Activity Code Charged starts with 4. Then, the Activity Code must end with {0-7} for the company code =01.

Then the application applies this validation rule:

Activity Code must end with {0-7} for the company code =01

This illustration shows how the application evaluates the condition filter and later applies the validation filter accordingly for the cross-validation rule.



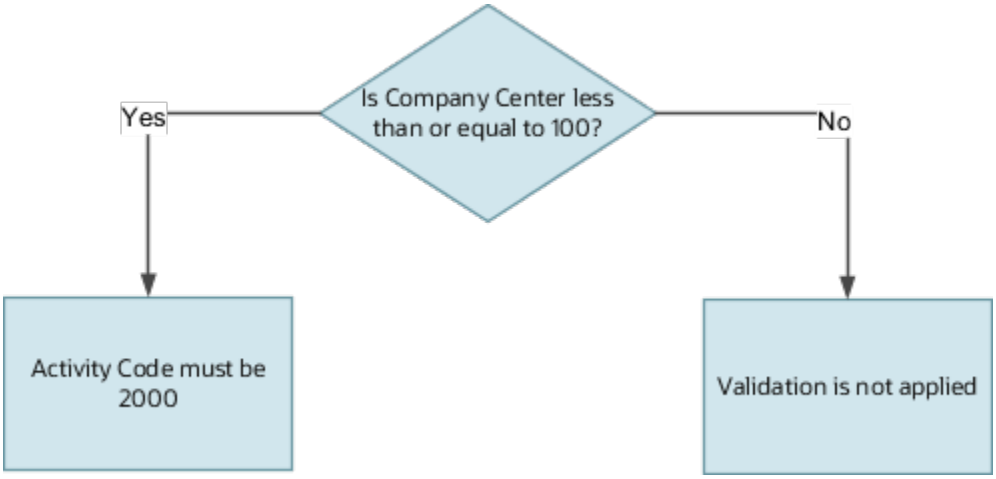
Use this CostKffValidationRule.dat file to apply the condition and validation filters:

```
METADATA|CostKffValidationRule|CrossValidationRuleId|StructureInstanceCode|RuleCode|RuleDescription|
ErrorMessageText|EnabledFlag|StartDateActive|EndDateActive|ConditionFilterText|ValidationFilterText
MERGE|CostKffValidationRule|32|HDL_COST_CVR_COST_KFF_INST|<RuleCode>|<ruleDesc>|<errorMsgTxt>||Y|2010/01/01|
4712/12/31|Company = 01,ACTIVITY_CODE STARTSWITH 3, ACTIVITY_CODE STARTSWITH 4|ACTIVITY_CODE ENDSWITH
 0,ACTIVITY_CODE ENDSWITH 1,ACTIVITY_CODE ENDSWITH 2,ACTIVITY_CODE ENDSWITH 3,ACTIVITY_CODE ENDSWITH
 4,ACTIVITY_CODE ENDSWITH 5,ACTIVITY_CODE ENDSWITH 6,ACTIVITY_CODE ENDSWITH 7
```

Example 3: Activity Code Based on Cost Center

If Cost Center is less than or equal to 100, then Activity Code must be 2000. The application evaluates the condition filter that the cost center is less than or equal to 100. If the condition filter is satisfied, then the application applies the validation filter that the activity code must be 2000.

This illustration shows how the application evaluates the condition filter and later applies the validation filter accordingly for the cross-validation rule:



Use this CostKffValidationRule.dat file to apply the condition and validation filters:

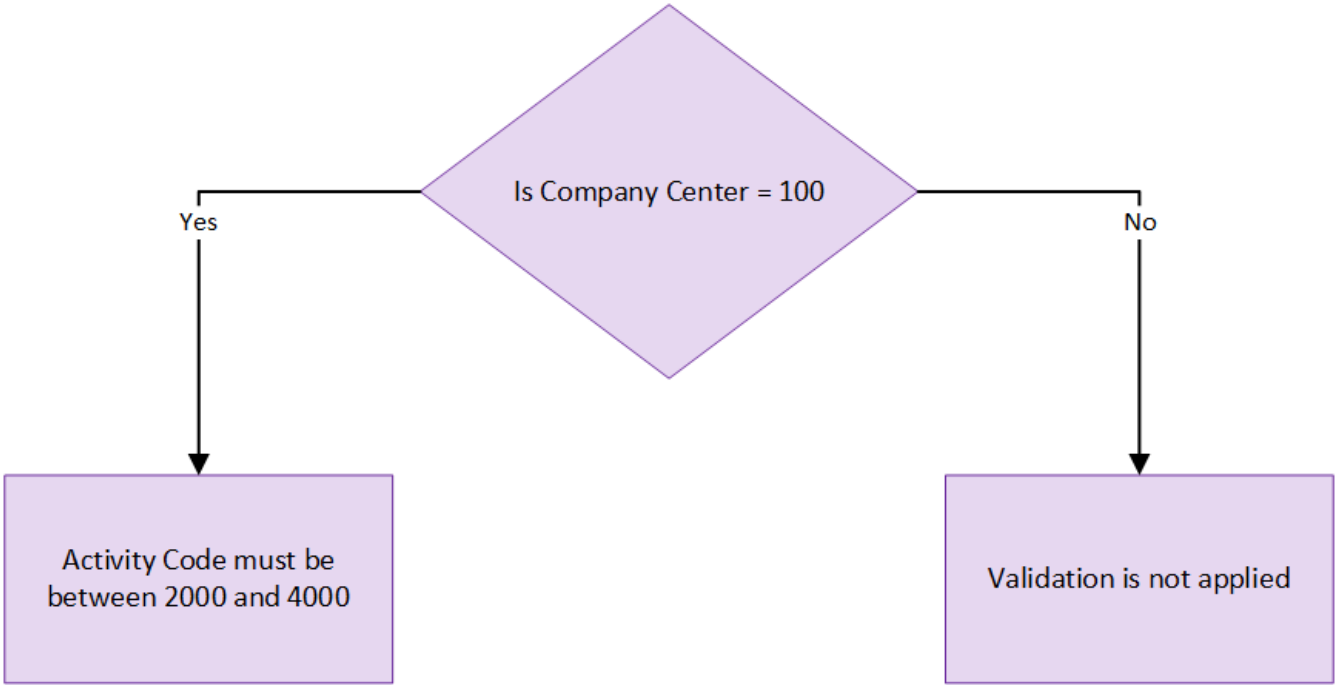
```
METADATA|CostKffValidationRule|StructureInstanceCode|RuleCode|RuleDescription|ErrorMessageText|EnabledFlag|
StartDateActive|EndDateActive|ConditionFilterText|ValidationFilterText
MERGE|CostKffValidationRule|33|HDL_COST_CVR_COST_KFF_INST|ACTIVITY_CODE_CHECK_RULE|Check Activity Code|
Activity Code must be 2000 for Cost Center less than 100.|Y|2010/01/01|4712/12/31|COST_CENTER <= 100|
ACTIVITY_CODE = 2000
```

Example 4: Activity Code Range for a specific Cost Center

If Cost Center is 100, then Activity Code must be between 2000 and 4000. The application evaluates the condition filter that the cost center is 100 or not. If the condition filter is satisfied, then the application applies the validation filter that the activity code must be between 2000 and 4000.

Note: No space between hyphen and the values in BETWEEN operator. (Ex: BETWEEN 2000-4000)

This illustration shows how the application evaluates the condition filter and later applies the validation filter accordingly for the cross-validation rule:



Use this.dat file to apply the condition and validation filters:

```
METADATA|CostKffValidationRule|StructureInstanceId|StructureInstanceCode|RuleC
ode|RuleDescription|ErrorMessageText|EnabledFlag|StartDateActive|EndDateActive
|ConditionFilterText|ValidationFilterText
MERGE|CostKffValidationRule||HDL_COST_CVR_COST_KFF_INST|ACTIVITY_CODE_BETWEEN_
CHECK|Activity Code between validation for Cost Center|Activity Code must be
between 2000 and 4000 if Cost Center is 100|Y|2018/01/01||COST_CENTER =
100|ACTIVITY_CODE BETWEEN 2000-4000
```

Related Topics

- Overview of Loading Costing Cross-Validation Rules
- Examples of Updating Costing Cross-Validation Rules

Examples of Updating Costing Cross-Validation Rules

As a best practice, you must define and enable the cross-validation rules before you run the costing process for the first time.

After you set the end date for your validation rule, you must also set the end date for the existing combinations with that rule.

You can use **Cross Validation Rule Report.catalog**, uploaded in Cloud Customer Connect, to find all the existing cross validation rules.

Delete Validation Rules

Using HCM Data Loader, you can delete validation rules that you erroneously created.

```
METADATA|CostKffValidationRule|StructureInstanceCode|RuleCode
DELETE|CostKffValidationRule|HDL_COST_CVR_COST_KFF_INST|ACTIVITY_CHECK_RULE
```

End Date Validation Rules

Use HCM Data Loader to set the end date for your validation rules. Let's consider this example. There's an existing validation rule that states that the Activity Code is Prohibited for Company = 01, which isn't valid any more. You want to set the end date for this rule to 31-Jul-2019.

Use this dat file to set the end date for your validation rule.

```
METADATA|CostKffValidationRule|StructureInstanceCode|RuleCode|StartDateActive|EndDateActive
MERGE|CostKffValidationRule|HDL_COST_CVR_COST_KFF_INST|ACTIVITY_RULE_CODE|2020/12/31
```

Enable or Disable Validation Rules

You can use HCM Data Loader to disable the existing validation rule.

Example: An existing validation rule states that if Activity Code is equal to 2200, then Location can't be equal to 2201 and 22EE. You want to disable this rule because you don't want to validate the rules in earlier periods.

Use this dat file to disable validation rules.

```
METADATA|CostKffValidationRule|CrossValidationRuleId|StructureInstanceCode|RuleCode|RuleDescription|
ErrorMessageText|EnabledFlag|StartDateActive|EndDateActive|ConditionFilterText|ValidationFilterText
MERGE|CostKffValidationRule|31|HDL_COST_CVR_COST_KFF_INST|<RuleCode>|<ruleDesc>|<errorMsgTxt|Y|2010/01/01|
4712/12/31|Company = 01,ACTIVITY_CODE ISNOTBLANK|ACTIVITY_CODE STARTSWITH 0,ACTIVITY_CODE STARTSWITH
 1,ACTIVITY_CODE STARTSWITH 2,ACTIVITY_CODE STARTSWITH 3,ACTIVITY_CODE STARTSWITH 4
```

In this example, you update the existing rule to extend it for additional conditions. You have an existing validation rule that states that Cost Center is required for Company=01. In this rule, blank isn't a valid value. You want to extend this rule for Company =02.

Use this dat file to extend the validation rule for Company =02.

```
METADATA|CostKffValidationRule|StructureInstanceCode|RuleCode|RuleDescription|ErrorMessageText|EnabledFlag|
StartDateActive|EndDateActive|ConditionFilterText|ValidationFilterText
MERGE|CostKffValidationRule|HDL_COST_CVR_COST_KFF_INST|<RuleCode>|<ruleDesc>|<errorMsgTxt>|Y|2010/01/01|
4712/12/31|Company = 01,Company = 02|COST_CENTER ISNOTBLANK
```

In another scenario, you might want to extend this rule such that Cost Center and Activity Code are required for Company = 01 and 02.

Use this dat file to extend the validation rule for Activity Code.

```
METADATA|CostKffValidationRule|StructureInstanceCode|RuleCode|RuleDescription|ErrorMessageText|EnabledFlag|
StartDateActive|EndDateActive|ConditionFilterText|ValidationFilterText
MERGE|CostKffValidationRule|HDL_COST_CVR_COST_KFF_INST|<RuleCode>|<ruleDesc>|<errorMsgTxt>|Y|2010/01/01|
4712/12/31|Company = 01|COST_CENTER ISNOTBLANK,ACTIVITY_CODE ISNOTBLANK
```

Related Topics

- Overview of Loading Costing Cross-Validation Rules
- Examples of Updating Costing Cross-Validation Rules

Example of Updating Cost Code Combinations

You can use HCM Data Loader to end date or disable the existing cost account combinations so that the invalid cost results doesn't get created based on the new organization policy.

For example, your organization has determined that the company Operations can't use the cost center Marketing. However, you already have the cost account combinations with this data. If you don't set the end date or disable these combinations, the costing process will continue to refer to the existing combinations and successfully create the costing results.

You can use **Cost Combination Report.catalog**, uploaded in Cloud Customer Connect at https://cloudcustomerconnect.oracle.com/posts/0b2fb95bc4, to find all the existing cost code combinations.

Let's consider these examples.

Example 1

As per the new organization rule effective on 01-Aug-2019, the Funding Source must not be 1010 for the Fund 4310. However, the cost code combinations already exists with these values. So, you end date all the combinations that have Fund = 4310 and Funding Source = 1010 with 31-Jul-2019.

Use this cost account combination.dat file to end date the cost account combinations. This dat file is created with a single record that has the combination. You need to add all the records that has this segment combination. Once you set an end date to all the existing cost account combinations with this data, the application gives an error message, saying that "FLEX-COMBINATION HAS EXPIRED". If you have setup suspense account at payroll or department level, then the costing result is created with suspense account. If there's no suspense account, the payroll process will error.

You must provide CostAllocationKeyflexId for performing this operation.

```
METADATA|CostAccountCombination|CostAllocationKeyflexId|EndDateActive|IdFlexNum
MERGE|CostAccountCombination|300100032377007|2019/07/31|448
```

Example 2

As per the new organization rule, the Department 3710 must not use Program as 6530, including prior periods. In this case, you can use HCM Data Loader to disable all cost code combinations with these values. Once you disable all the existing cost account combinations with this data, the application gives an error message, saying that "FLEX-COMBINATION DISABLED". If you have setup suspense account at payroll or department level, then the costing result gets created with suspense account. If there's no suspense account, the payroll process will error.

This dat file is created with a single record that has the combination. You need to add all the records that has this segment combination.

You must provide CostAllocationKeyflexId for performing this operation.

```
METADATA||CostAccountCombination|CostAllocationKeyflexId|EnabledFlag|IdFlexNum
MERGE|CostAccountCombination|300100032377007|N|448
```

Related Topics

- Overview of Loading Costing Cross-Validation Rules

20 Loading Payroll Data Using Transformation Formula

Payroll Transformation Formula for HCM Data Loader

Overview

Often times, your existing data or the payroll data that you upload might not be in the format recognized by HCM Data Loader.

In such cases, you can use a payroll transformation formula to transform your data into a format that'ssupported by HCM Data Loader.

Let's consider these examples.

- An inbound file contains data that needs to be loaded using different payroll business objects in HCM Data Loader. Here, the content of the file needs to be split across more than one HCM Data Loader file.
- You might create a transformation formula to convert an attribute value in the file to another value that you derive using value sets.
- You want to change a person number into an assignment number. In this case, you will use a more complex formula to convert the attributes.

You use the **Load Data From File** flow to transform your data into the HCM Data Loader file format using your transformation formula.

As this table shows, the two flow patterns are secured using these privileges:

Flow Pattern	Privileges
Submit Payroll Flow	PAY_SUBMIT_PAYROLL_FLOW_PRIV
Load HCM Data	HRC_LOAD_HCM_DATA_PRIV

This example specifies the file name in the formula as `PersonalPaymentMethod`, the file discriminator as `PersonalPaymentMethod`, and the business operation as MERGE.

```
/*HDL Related Outputs*/
 FileName = 'PersonalPaymentMethod'
 BusinessOperation = 'MERGE'
 FileDiscriminator = 'PersonalPaymentMethod'
```

To view details about the file name, file discriminator, and a list of supported business operations, use the **View Business Objects** task in the Data Exchange work area.

1. On the View Business Objects page, search for and select your business object. In this example, the business object is Personal Payment Method.

2. On the Component Details page, you can find the name of the file, and the file discriminator and a list of supported actions for the object.

Related Topics

- How You Transform Data Using Payroll Transformation Formula for HCM Data Loader
- How To Create A Program for Automation
- Submit the Load Data From File Flow

How You Transform Data Using Payroll Transformation Formula for HCM Data Loader

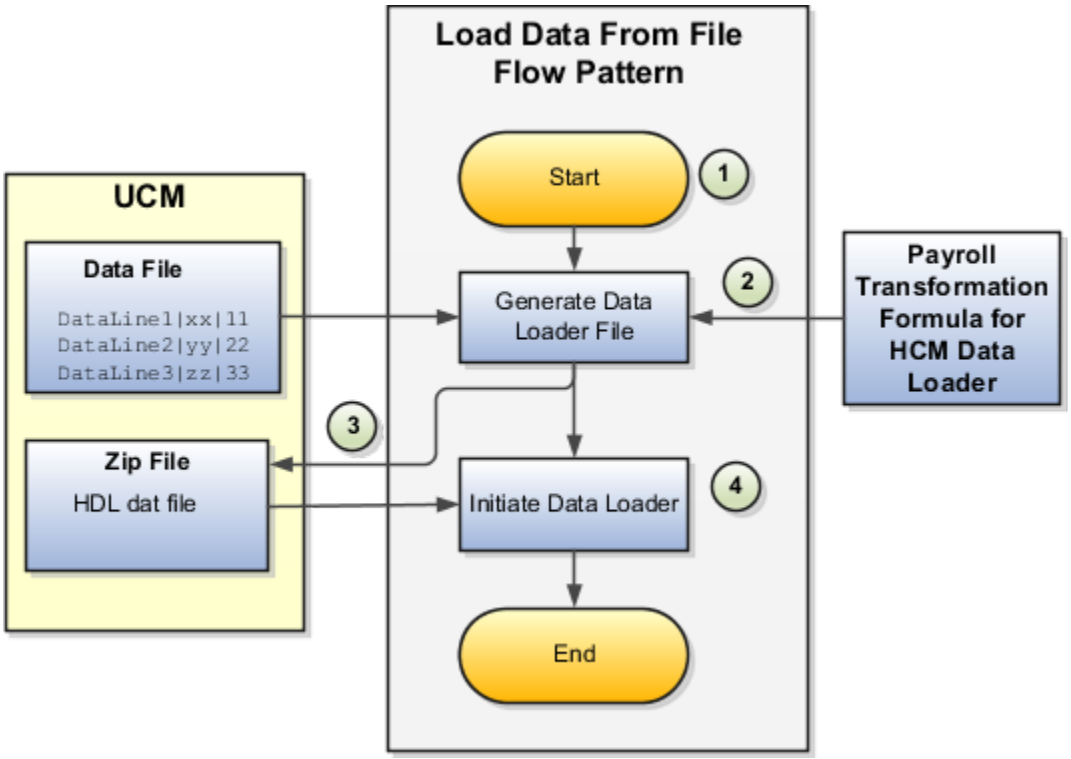
Use the **Load Data From File** flow to transform data in the source file into a format that's supported by HCM Data Loader.

You can submit this flow independently or include it in a flow that you create for automating data loads on aperiodic basis. When you submit the flow, either manually or using a web service, you must specify a transformationformula to transform the data, as needed.

The flow contains these two tasks that help you to transform data into a HCM Data Loader format:

- Generate Data Loader File
- Initiate Data Loader Task

As this figure shows, the first step is to submit the **Load Data From File** flow. This flow takes data from the flat file and generates an equivalent file format for the data present in the input file.



Perform these steps to transform data using Payroll Transformation Formula for HCM Data Loader:

1. On the Home page, click the **Submit a Flow** quick action under the **My Clients Groups** tab. On the Flow Submission page, search for, select, and submit the **Load Data From File** flow pattern.
2. The flow invokes the Payroll Transformation Formula for the Content ID. Typically, you create your transformation formula for HCM Data Loader on the Manage Fast Formulas page. The type of the formula should be HCM Data Loader.
3. The Generate Data Loader File task reads the data file line by line, producing an equivalent HCM Data Loader format for each line. Finally, it creates a compressed file of all of the transformed data files and uploads to the Oracle WebCenter Content server. Also, the task records the Content ID.
4. The **Initiate Data Loader** task takes the Content ID for the file generated by the **Generate Data Loader File** task. And it invokes HCM Data Loader. HCM Data Loader validates the data and creates valid records in the HCM cloud.

   The table shows the tasks and the privileges that they're secured with:

Task	Privilege
Submit Payroll Flow	PAY_SUBMIT_PAYROLL_FLOW_PRIV
Load HCM Data	HRC_LOAD_HCM_DATA_PRIV

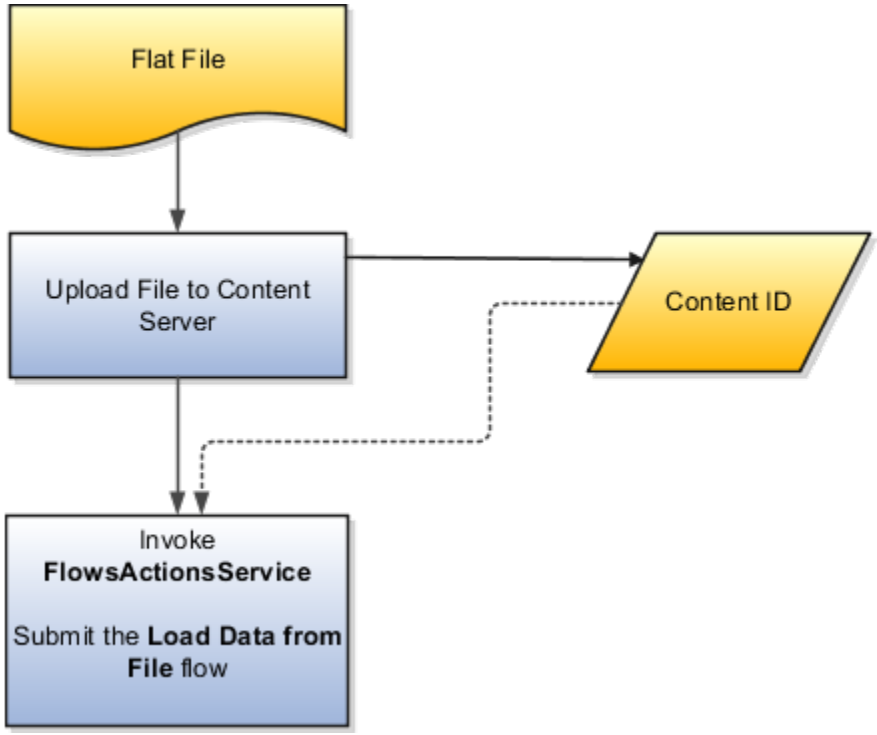
Related Topics

- How To Create A Program for Automation
- Submit the Load Data From File Flow
- Transformation Formula Input Variables

How To Create A Program for Automation

You can submit the **Load Data from File** flow by using a web service.

As this figure shows, your program uploads the source file to content server, and retrieves the content ID for the Flow Actions Service web service. Then, the program calls the Flow Actions web service by supplying certain parameters.



When calling the web service, your program supplies these parameters.

- Name of the flow pattern, which is **Load Data from File**
- Content ID of the uploaded file
- Unique name to identify the flow instance being submitted
- Process configuration group ID for special processing (optional)
- Transformation formula ID (mandatory)

For more information about the Flow Actions Service web service, refer to the SOAP Web Services for Oracle HCM Cloud guide. For examples of its usage for automating file uploads, refer to the attachment for HCM Data Loader User Guide (1664133.1) on My Oracle Support at https://support.oracle.com.

Related Topics

- Submit the Load Data From File Flow
- Transformation Formula Input Variables
- Payroll Transformation Formula Operations
- Sample Payroll Transformation Formula for HCM Data Loader
- Sample Payroll Transformation Formula for Multiple Business Objects

Submit the Load Data From File Flow

On the Home page, click the **Submit a Flow** quick action under the My Clients Groups tab. On the Flow Submission page, submit the **Load Data from File** flow pattern to transform data in your source file into the HDL format.

Assumptions

This procedure has these assumptions.

- You have the Human Capital Management Integration Specialist role.
- You have the source file ready to upload to Oracle WebCenter Content.
- If you have already uploaded the source file, you have the content ID handy.

Note: To upload files to the content server, browse to the source file on your file system, check it in to the content server, and retrieve its Content ID. For more information, see Oracle Fusion Middleware Using Oracle WebCenter Content guide.

Before You Begin

Before you submit the flow, ensure that you meet these prerequisites.

Characteristics	What You Should Do
Data transformation	If the data in the source file requires transformation, create and compile a transformation formula using the HCM Data Loader formula type. On the Home page, click the **Fast Formulas** quick action under the **My Clients Group** tab. You can now specify the processing parameters for your process configuration group.
File encryption	Before loading encrypted files, ensure that the encryption keys exist for the secure file transfer. This process involves creating a service request, generating PGP key pairs, and sharing the encryption keys. Specify the **Payroll Batch Loader Encryption Type** parameter value for the process configuration group you select when running the flow or the web service. Valid values are `PGPSIGNED`, `PGPUNSIGNED`, and `NONE`. On the Home page, click the **Process Payroll Configuration** quick action under the My Clients group tab. Now, you can specify the processing parameters for your process configuration group.
Other processing parameters	Use the **Payroll Process Configuration** task to add parameters for the process configuration group. Examples of processing parameters include Batch Error Mode, Logging Area, Logging Category, and Threads.

1. On the Home page, click the **Submit a Payroll Flow** quick action under the **My Clients Groups** tab.
2. In the **Legislative Data Group** option, select a legislative data group.
3. Search for and select the **Load Data from File** flow pattern.
4. Click **Next**.
5. Enter the parameters, as shown in this table.

Field	Value
Payroll Flow	Descriptive name for this specific flow process.

Field	Value
Content Id	Enter the Content Id. The source file must already exist on the content server.
Transformation Formula	Select the required transformation formula. The type of the formula should be **HCM Data Loader**.
Process Configuration Group	Select your process configuration group.

6. On the Enter Parameters page, click **Next**.
7. On the Enter Flow Interaction page, click **Next**.
8. On the Schedule page, click **Next**.
9. On the Review page, click **Submit**.
10. In the confirmation dialog box, click **OK and View Checklist**.
11. On the Payroll Flow page, Task Details tab you should see a green check mark in the **Generate Data Loader File** and **Initiate Data Loader** rows, Task Type column. If not, on the toolbar, click the **Refresh** icon intermittently until you do.
12. Close the Payroll Flow page.
13. On the Overview page, search for and click your payroll flow.
14. View the process results.
15. Check for any errors or warnings.

Related Topics

- Transformation Formula Input Variables
- Payroll Transformation Formula Operations
- Return Values for Payroll Transformation Formula for HCM Data Loader
- Sample Payroll Transformation Formula for HCM Data Loader
- Sample Payroll Transformation Formula for Multiple Business Objects

Transformation Formula Input Variables

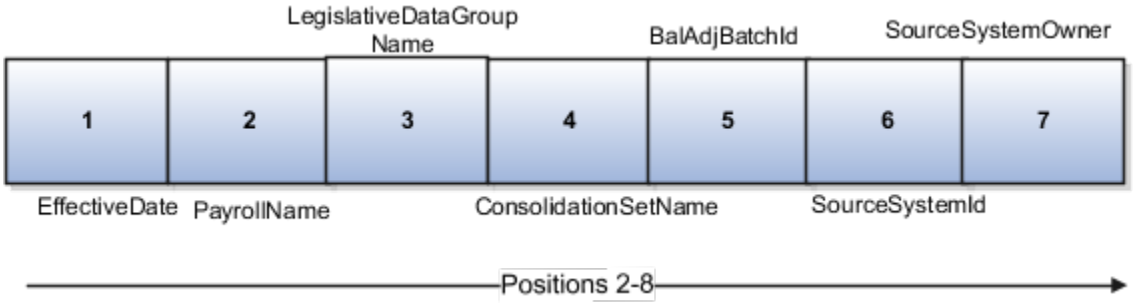
Variables, such as `FileName`, `FileDiscriminator` and `LINEREPEATNO`, are available for all formulas of HCM Data Loader Transformation formula type. Additional variables may be available depending on the selected business object.

Positions

Using the transformation formula, you can assign attributes to the required position. Positions can range from 1 to N. Depending upon the business object, the positions can be either optional or mandatory.

Example:

This figure shows the different attributes for positions 2 through 8 for Balance Adjustments.



In this example, you assign attributes to positions 2 through 8.

- `POSITION2`: `EffectiveDate`
- `POSITION3`: `PayrollName`
- `POSITION4`: `LegislativeDataGroupName`
- `POSITION5`: `ConsolidationSetName`
- `POSITION6`: `BalAdjBatchId`
- `POSITION7`: `SourceSystemId`
- `POSITION8`: `SourceSystemOwner`

FileName, FileDiscriminator, and BusinessOperation

`FileName`, `FileDiscriminator`, and `BusinessOperation` variables are required for all transformations.

Here are the details of these variables.

- `FileName` is the name of the file for the business object.
- `FileDiscriminator` is the file discriminator for the business object.
- `BusinessOperation` refers to the operation, such as Merge or Delete that are performed by the HCM Data Loader process on the transformed file.

Here's an example of values that you can supply for the input variables: `FileName`, `FileDiscriminator`, and `BusinessOperation`.

```
FileName = 'BalanceAdjustmentHeader'
BusinessOperation = 'MERGE'
FileDiscriminator = POSITION1
```

LINEREPEAT And LINEREPEATNO

`LINEREPEAT` allows a single line of input to be processed multiple times. And `LINEREPEATNO` indicates the number of repetitions.

For example, for time entry, there might be a regular time entry wage followed by a premium time entry wage.

Example: The Element Entry file contains these details.

```
Update|ElementEntryValue|Vision Corporation US LDG|WLM_Salary|2019/04/15|4712/12/31|E955160008191355-2|
Amount|1002|2|E
Update|ElementEntryValue|Vision Corporation US LDG|WLM_Salary|2019/04/15|4712/12/31|E955160008191355-2|
Amount|1003|3|E
```

The input line can be processed twice. The output file contains the element entry and element entry value as shown in this sample code snippet.

```
ELSE IF OPERATION='MAP' THEN
(
 LegislativeDataGroupName=POSITION3
 ElementName=POSITION4
 EffectiveStartDate=POSITION5
 EffectiveEndDate=POSITION6
 AssignmentNumber=POSITION7
 InputValueName=POSITION8
 ScreenEntryValue=POSITION9
 MultipleEntryCount=POSITION10
 EntryType=POSITION11
IF LINEREPEATNO=1 THEN
 (
 BusinessOperation='MERGE'
 BusinessObject='Element Entry'
 FileName = 'ElementEntry'
 FileDiscriminator ='ElementEntry'
 LINEREPEAT = 'Y'
 RETURN BusinessOperation,FileDiscriminator,FileName
 )
ELSE
(
 BusinessOperation='MERGE'
 BusinessObject='Element Entry Value'
 FileName = 'ElementEntry'
 FileDiscriminator = 'ElementEntry'
 LINEREPEAT = 'N'
 RETURN BusinessOperation,FileDiscriminator,FileName
 )
```

Note:

- The length of a line in the incoming raw file can't be more than 1000 characters
- The length of an attribute between two delimiters can't be more than 255 characters

Related Topics

- Payroll Transformation Formula Operations
- Return Values for Payroll Transformation Formula for HCM Data Loader
- Sample Payroll Transformation Formula for HCM Data Loader
- Sample Payroll Transformation Formula for Multiple Business Objects

Payroll Transformation Formula Operations

The transformation formula is invoked several times to derive different components that are required for processing the incoming data.

This table explains the various operations that you can do with the formula.

Operation Type	Return Value	Example
FILETYPE	OUTPUTVALUE	DELIMITED

Operation Type	Return Value	Example
DELIMITER	OUTPUTVALUE	|
READ	OUTPUTVALUE	NONE
MAP	Business Object Attributes	NA
NUMBEROFBUSINESSOBJECTS	OUTPUTVALUE	2
METADATALINEINFORMATION	METADATAn	METADATA1

Note: Remember that only 'Delimited' file type is supported.

Delimiter

The default delimiter that separates values is a pipe character. If your file uses a different delimiter, you must set the delimiter you want your formula.

This example specifies a comma character as the delimiter.

```
/* Calculations */
IF OPERATION='FILETYPE' THEN
 OUTPUTVALUE='DELIMITED'
ELSE IF OPERATION='DELIMITER' THEN
 OUTPUTVALUE=','
```

Note: Ensure that the delimiter you enter in the formula is a single non-ASCII character and not part of any of the values to upload.

MAP

The MAP operation defines the return values related to a particular object. The return values must have the same names as the attributes specified in the application for that object.

For example, these RETURN values can be used to generate an Element Entry dat file.

```
RETURN
 BusinessOperation,FileDiscriminator,FileName,AssignmentId,AssignmentNumber,CreatorType,DateEarned,EffectiveEndDate,
```

Here, the `BusinessOperation` is set to MERGE and the `BusinessObject` is set to Element Entry.

METADATALINEINFORMATION

The application generates the file either with all defined attributes or with only specified attributes, depending on whether you specify a value for `METADATALINEINFORMATION` or not.

1. If you don't specify a value for `METADATALINEINFORMATION`, then the application generates `METADATLINE` in the transformed file with all defined attributes for the business object.
2. If you specify a value, then the transformed file will contain only the attributes that you specified.

For the METADATALINEINFORMATION operation, you specify an array per business object being processed in the formula. The number of arrays should match the number specified in the NUMBEROFBUSINESSOBJECTS operation. The name of the array should be METADATA with the number as suffix . For example, RETURN METADATA1, METADATA2 when the NUMBEROFBUSINESSOBJECTS is 2.

Note: The first two entries in the array are reserved to specify the FileName and FileDiscriminator of the business object.

Additionally, for METADATALINEINFORMATION, you can specify attributes with special characters for that business objects.

Notice that in this example BalAdjBatchId(SourceSystemId) has parenthesis.

If the file contains either Flexfield or SourceSystem references, then the application can't resolve the default mapping of output parameter names and attributes.

Let's consider this syntax: jobEffSegment1(PER_JOBS_EIT_EFF=context). To allow this construct to be generated in the HCM Data Loader file, you define the METADATA line in the transformation formula. For each business object that appears in the output, you must define the METADATA content in an array.

Example:

```
METADATA2[1] = 'Job' /*FileName*/
METADATA2[2] = 'JobExtraInfo' /*FileDiscriminator*/
METADATA2[3] = 'EffectiveStartDate'
METADATA2[4] = 'EffectiveEndDate'
METADATA2[5] = 'JobCode'
METADATA2[6] = 'SetCode'
METADATA2[7] = 'FLEX:PER_JOBS_EIT_EFF'
METADATA2[8] = 'EFF_CATEGORY_CODE'
METADATA2[9] = 'InformationType'
METADATA2[10] = 'JeiInformationCategory'
METADATA2[11] = 'LegislationCode'
METADATA2[12]= 'SequenceNumber'
METADATA2[13]= 'jobEffSegment1(PER_JOBS_EIT_EFF=job-eff-context)'
```

Here's how the generated HCM Data Loader file looks like.

```
METADATA|JobExtraInfo|EffectiveStartDate|EffectiveEndDate|JobCode|SetCode|FLEX:PER_JOBS_EIT_EFF|
EFF_CATEGORY_CODE|InformationType|JeiInformationCategory|LegislationCode|SequenceNumber|
jobEffSegment1(PER_JOBS_EIT_EFF=job-eff-context)
```

NUMBEROFBUSINESSOBJECTS

This operation indicates the number of business objects being processed in the formula.

Related Topics

- Return Values for Payroll Transformation Formula for HCM Data Loader
- Sample Payroll Transformation Formula for HCM Data Loader
- Sample Payroll Transformation Formula for Multiple Business Objects

Return Values for Payroll Transformation Formula for HCM Data Loader

The return values for HCM Data loader formulas vary based on the business object and task action.

They are the same as the attribute names and must include `BusinessOperation`, `FileName`, and `FileDiscriminator`.

Here's an example of return values.

```
/*Return Values*/
RETURN BusinessOperation,FileName,FileDiscriminator,Attribute1,Attribute2,...,Attributen
/*Attributes for a particular Business Object can be found from the View Business Objects UI under the HCM
 Data Loader task in the Data Exchange Work Area */
```

For the `NUMBEROFBUSINESSOBJECTS` and `METADATALINEINFORMATION` operations, the `RETURN` statement is as follows.

```
/*Return Values for NUMBEROFBUSINESSOBJECTS and METADATALINEINFORMATION Operation*/
IF OPERATION='FILETYPE' THEN
 OUTPUTVALUE='DELIMITED'
ELSE IF OPERATION='DELIMITER' THEN
 OUTPUTVALUE='|'
ELSE IF OPERATION='READ' THEN
 OUTPUTVALUE='NONE'
ELSE IF OPERATION = 'NUMBEROFBUSINESSOBJECTS' THEN(
 OUTPUTVALUE = '2'
 RETURN OUTPUTVALUE
 )
ELSE IF OPERATION = 'METADATALINEINFORMATION' THEN
 (
 METADATA1[1] = 'BalanceAdjustmentHeader' /*FileName*/ /*Reserved*/
 METADATA1[2] = 'BalanceAdjustmentHeader' /*FileDiscriminator*/ /*Reserved*/
METADATA1[3] = 'LegislativeDataGroupName'
 METADATA1[4] = 'BatchName'
 METADATA1[5] = 'SourceSystemId'
 METADATA1[6] = 'SourceSystemOwner'

 METADATA2[1] = 'BalanceAdjustmentHeader' /*FileName*/ /*Reserved*/
 METADATA2[2] = 'BalanceAdjustmentGroup' /*FileDiscriminator*/ /*Reserved*/
 METADATA2[3] = 'EffectiveDate'
 METADATA2[4] = 'PayrollName'
 METADATA2[5] = 'LegislativeDataGroupName'
 METADATA2[6] = 'ConsolidationSetName'
 METADATA2[7] = 'BalAdjBatchId(SourceSystemId)'
 METADATA2[8] = 'SourceSystemId'
 METADATA2[9] = 'SourceSystemOwner'
 RETURN METADATA1, METADATA2 /*Only two as Return value for NUMBEROFBUSINESSOBJECTS is 2*/
 )
```

You can define variables with special characters. For example, use this structure to return the `BalAdjBatchId(SourceSystemId)` Source ID.

```
/*Return Values for the MAP Operation*/
 FileName = 'BalanceAdjustmentHeader'
 BusinessOperation = 'MERGE'
 FileDiscriminator = POSITION1
 EffectiveDate = POSITION2
 PayrollName = POSITION3
 LegislativeDataGroupName = POSITION4
 ConsolidationSetName = POSITION5
 "BalAdjBatchId(SourceSystemId)"= POSITION6
 SourceSystemId = POSITION7
 SourceSystemOwner = POSITION8

 RETURN BusinessOperation, FileDiscriminator, FileName, EffectiveDate, PayrollName,
 LegislativeDataGroupName, ConsolidationSetName, "BalAdjBatchId(SourceSystemId)", SourceSystemId,
 SourceSystemOwner

/*Note BalAdjBatchId(SourceSystemId) is enclosed by double quotes while assigning value as well as while
 putting it in the return values list */
```

Related Topics

- Overview of Payroll Transformation Formula For HCM Data Loader
- How You Transform Data Using Payroll Transformation Formula for HCM Data Loader
- Sample Payroll Transformation Formula for HCM Data Loader
- Sample Payroll Transformation Formula for Multiple Business Objects

Sample Payroll Transformation Formula for Multiple Business Objects

In this example, the formula uses the user defined tables and personal payment method business objects. It converts the Person Number in the flat file into Assignment Number and uses the `METADATALINEINFORMATION` and `NUMBEROFBUSINESSOBJECTS` operations.

Here's the sample of the raw input file.

```
PPM|2018/04/04|1|ZHRX_VS_US_TPPI_LDG_ONE|955160008191423|ZHRX_VS_US_TPPI_Check|PPM1|M|10
UDT|SM_UDT_4|Range|Number|Test UDT|USA LDG
```

And this code snippet has the formula for this example.

```
/*************************************************************
FORMULA NAME: Load User Defined Table and Personal Payment Method
FORMULA TYPE: HCM Data Loader
*************************************************************/
/* Inputs */
INPUTS ARE OPERATION (text), LINENO (number), LINEREPEATNO (number),POSITION1 (text), POSITION2 (text),
 POSITION3 (text), POSITION4 (text), POSITION5 (text), POSITION6 (text), POSITION7 (text), POSITION8
 (text),POSITION9 (text)
DEFAULT FOR POSITION1 IS 'NO DATA'
DEFAULT FOR POSITION2 IS 'NO DATA'
DEFAULT FOR POSITION3 IS 'NO DATA'
DEFAULT FOR POSITION4 IS '2'
DEFAULT FOR POSITION5 IS '100'
DEFAULT FOR POSITION6 IS 'NO DATA'
DEFAULT FOR POSITION7 IS 'NO DATA'
DEFAULT FOR POSITION8 IS 'NO DATA'
DEFAULT FOR POSITION9 IS 'NO DATA'
DEFAULT FOR LINEREPEATNO IS 1
IF OPERATION='FILETYPE' THEN
 OUTPUTVALUE='DELIMITED'
ELSE IF OPERATION='DELIMITER' THEN
 OUTPUTVALUE='|'
ELSE IF OPERATION='READ' THEN
 OUTPUTVALUE='NONE'
ELSE IF OPERATION = 'NUMBEROFBUSINESSOBJECTS' THEN
 (
 OUTPUTVALUE = '2'
 RETURN OUTPUTVALUE
 )
ELSE IF OPERATION = 'METADATALINEINFORMATION' THEN
 (
 METADATA1[1] = 'UserDefinedTable' /*FileName*/
 METADATA1[2] = 'UserDefinedTable' /*FileDiscriminator*/
 METADATA1[3] = 'UserTableCode'
 METADATA1[4] = 'RangeOrMatch'
 METADATA1[5] = 'UserKeyUnits'
 METADATA1[6] = 'UserRowTitle'
```

```
 METADATA1[7] = 'UserTableName'
 METADATA1[8] = 'LegislativeDataGroupName'
 METADATA2[1] = 'PersonalPaymentMethod' /*FileName*/
 METADATA2[2] = 'PersonalPaymentMethod' /*FileDiscriminator*/
 METADATA2[3] = 'EffectiveStartDate'
 METADATA2[4] = 'PersonalPaymentMethodCode'
 METADATA2[5] = 'AssignmentNumber'
 METADATA2[6] = 'Amount'
 METADATA2[7] = 'ProcessingOrder'
 METADATA2[8] = 'OrganizationPaymentMethodCode'
 METADATA2[9] = 'PaymentAmountType'
 METADATA2[10] = 'LegislativeDataGroupName'
 RETURN METADATA1, METADATA2
 )
ELSE IF OPERATION='MAP' THEN

 IF POSITION1='UDT' THEN
 (
 FileName = 'UserDefinedTable'
 BusinessOperation = 'MERGE'
 FileDiscriminator = 'UserDefinedTable'
 UserTableCode = POSITION2
 IF POSITION3='Range' THEN
 (
 RangeOrMatch = 'R'
 )
 IF POSITION4='Number' THEN
 (
 UserKeyUnits = 'N'
 )
 UserRowTitle = POSITION5
 UserTableName = POSITION2
 LegislativeDataGroupName = POSITION6
 RETURN
 BusinessOperation,FileDiscriminator,FileName,UserTableCode,RangeOrMatch,UserKeyUnits,UserRowTitle,UserTableName,Leg
 )
 IF POSITION1='PPM' THEN
 (
 FileName = 'PersonalPaymentMethod'
 BusinessOperation = 'MERGE'
 FileDiscriminator = 'PersonalPaymentMethod'
 EffectiveStartDate=POSITION2
 ProcessingOrder=POSITION3
 LegislativeDataGroupName=POSITION4
 AssignmentNumber=GET_VALUE_SET('SAMPLE_GET_ASG_NUM','|=PERSON_NUMBER='''||POSITION5||'''')
 OrganizationPaymentMethodCode=POSITION6
 PersonalPaymentMethodCode=POSITION7
 PaymentAmountType=POSITION8
 Amount=POSITION9
 RETURN
 BusinessOperation,FileName,FileDiscriminator,EffectiveStartDate,PersonalPaymentMethodCode,AssignmentNumber,Amount,P
 )
ELSE
 OUTPUTVALUE='NONE'
RETURN OUTPUTVALUE
/* End Formula Text */
```

Note: To debug value sets, create a BI report with this query to return the required data.

```
SELECT pay_ff_functions.gvs ('SAMPLE_GET_ASG_NUM','|=PERSON_ID=100000012092216') value FROM dual;
```

Related Topics

- Overview of Payroll Transformation Formula For HCM Data Loader
- How You Transform Data Using Payroll Transformation Formula for HCM Data Loader
- How To Create A Program for Automation

Sample Payroll Transformation Formula for HCM Data Loader

In this example, the transformation formula specifies the transformation mechanism for an incoming comma separated delimited file. The formula's return values are the same as the list of attributes for the personal payment method object.

Here's the sample raw file for personal payment method, with comma as the delimiter.

```
2018/04/04,1,ZHRX_VS_US_TPPI_LDG_ONE,E955160008191423,ZHRX_VS_US_TPPI_Check,PPM1,M,10
```

And this code snippet has the formula for this example.

```
/*****************************************************************
FORMULA NAME: Load Personal Payment Method
FORMULA TYPE: HCM Data Loader
*****************************************************************/
/* Inputs */
INPUTS ARE OPERATION (text), LINENO (number), LINEREPEATNO (number),POSITION1 (text), POSITION2 (text),
 POSITION3 (text), POSITION4 (text), POSITION5 (text), POSITION6 (text), POSITION7 (text), POSITION8 (text)

DEFAULT FOR POSITION1 IS 'NO DATA'
DEFAULT FOR POSITION2 IS 'NO DATA'
DEFAULT FOR POSITION3 IS 'NO DATA'
DEFAULT FOR POSITION4 IS '2'
DEFAULT FOR POSITION5 IS '100'
DEFAULT FOR POSITION6 IS 'NO DATA'
DEFAULT FOR POSITION7 IS 'NO DATA'
DEFAULT FOR POSITION8 IS 'NO DATA'
DEFAULT FOR LINEREPEATNO IS 1

IF OPERATION='FILETYPE' THEN
 OUTPUTVALUE='DELIMITED'
ELSE IF OPERATION='DELIMITER' THEN

(
 OUTPUTVALUE=','
 RETURN OUTPUTVALUE
)
ELSE IF OPERATION='READ' THEN
 OUTPUTVALUE='NONE'
ELSE IF OPERATION='MAP' THEN
 /*HDL Related Outputs*/
 (
 FileName = 'PersonalPaymentMethod'
 BusinessOperation = 'MERGE'
 FileDiscriminator = 'PersonalPaymentMethod'
 EffectiveStartDate=POSITION1
 ProcessingOrder=POSITION2
 LegislativeDataGroupName=POSITION3
 AssignmentNumber=POSITION4
 OrganizationPaymentMethodCode=POSITION5
 PersonalPaymentMethodCode=POSITION6
 PaymentAmountType=POSITION7
 Amount=POSITION8
```

```
 RETURN
 BusinessOperation,FileName,FileDiscriminator,EffectiveStartDate,PersonalPaymentMethodCode,AssignmentNumber,Amount,P
 )
ELSE
 OUTPUTVALUE='NONE'
RETURN OUTPUTVALUE
/* End Formula Text */
```

Related Topics

- Overview of Payroll Transformation Formula For HCM Data Loader
- How You Transform Data Using Payroll Transformation Formula for HCM Data Loader
- Sample Payroll Transformation Formula for Multiple Business Objects

SET Instructions in Transformation Formulas

You can configure the HCM Data Loader Transformation Formula to add SET instructions. On the HCM Data Loader DAT files, add the SET instructions before the METADATA instruction.

SET instructions are optional control commands that alter the default processing of the HDL file in which they appear.

Note: You can have multiple SET instructions.

Let's consider these examples.

Example 1

You can use the `ENABLE_INCREMENTAL_LOAD_EVENTS` set command to enable payroll events on the data that's loaded using SET instructions.

Here's the formula to enable the payroll events:

```
ELSE IF OPERATION = 'SETINSTRUCTIONS' THEN
(
SET1[1] = 'PersonalPaymentMethod' /*FileName*/
SET1[2] = 'ENABLE_INCREMENTAL_LOAD_EVENTS Y'
RETURN SET1
)
```

Example 2

Let's assume that you're loading personal payment method data and you want the name of the personal payment method to have a pipe. The name of the personal payment method should be PPM|1.

Here's the transformed HCM Data Loader file with a single line to load the personal payment method data with delimiter as pipe.

```
METADATA|PersonalPaymentMethod|EffectiveStartDate|PersonalPaymentMethodCode|AssignmentNumber|Percentage|
ProcessingOrder|OrganizationPaymentMethodCode|PaymentAmountType|LegislativeDataGroupName
MERGE|PersonalPaymentMethod|2018/04/04|PPM\|1|E300100006342154|100|999|PM US Sun Power Check |P|PM US Sun Po
```

Here's a sample snippet of the formula text to load personal payment method and user defined table:

```
ELSE IF OPERATION = 'SETINSTRUCTIONS' THEN
(
SET1[1] = 'UserDefinedTable' /*FileName*/
```

```
SET1[2] = 'FILE_ESCAPE \'
SET2[1] = 'PersonalPaymentMethod' /*FileName*/
SET2[2] = 'FILE_ESCAPE \'
RETURN SET1, SET2
)
```

Related Topics

- The SET Instruction

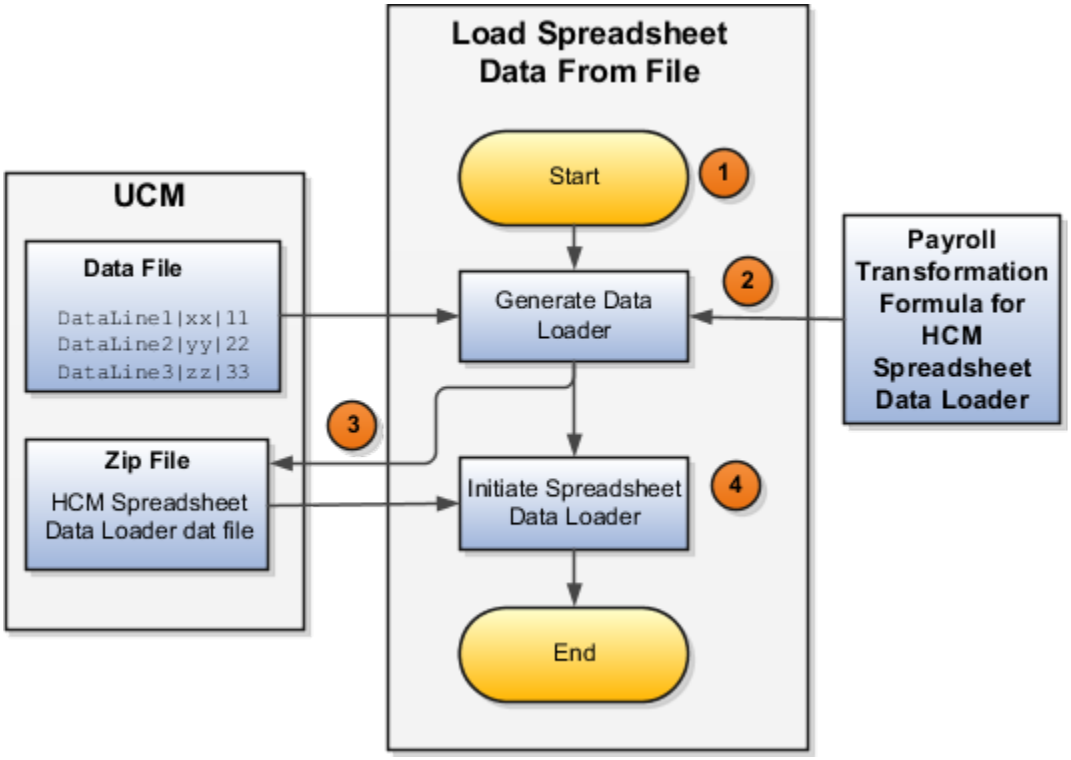
Payroll Transformation Formula for HCM Spreadsheet Data Loader

Overview

You can use HCM Spreadsheet Data Loader to load all payroll objects that HCM Data Loader supports.

As the first step,you create a spreadsheet template for the required object from the Data Exchange Work area and further downloadthe template in CSV format. You can download CSV and XML file templates from a spreadsheet template. The PayrollTransformation Formula for HCM Spreadsheet Data Loader transforms the raw delimited file to a format that suits the template.

This figure summarizes the process of transforming data that's uploaded using HCM Spreadsheet Data Loader.



Here's a summary of how the transformation process works.

1. On the Home page, click the **Submit a Flow** quick action under the **My Clients Groups** tab. Submit the **Load Spreadsheet Data File** flow pattern.

   The flow is secured using these privileges:

Flow	Privilege
Submit Payroll	`PAY_SUBMIT_PAYROLL_FLOW_PRIV`
Load Data using HCM Spreadsheet Data Loader	`HRC_LOAD_DATA_USING_HSDL_PRIV`

2. This flow pattern invokes the transformation formula for the Content ID and has these tasks:
   a. Generate Data Loader File
   b. Initiate Spreadsheet Data Loader
3. The **Generate Data Loader File** task reads the data file line by line, producing an equivalent HCM Spreadsheet Data Loader format for each line. Finally, it creates a compressed file of all of the transformed data files and uploads it to Oracle WebCenter Content server.
4. The **Initiate Spreadsheet Data Loader** task takes the compressed file generated by the **Generate Data Loader File** task and invokes the HCM Spreadsheet Data Loader. The HCM Spreadsheet Data Loader creates the required data in the HCM Cloud.

Related Topics

- Guidelines for Using HCM Spreadsheet Data Loader
- How Data Is Uploaded Using HCM Spreadsheet Data Loader
- HCM Spreadsheet Data Loader Templates
- Create and Edit Spreadsheet Templates
- Guidelines for Designing Spreadsheet Templates

Sample Payroll Transformation Formula for HCM Spreadsheet Data Loader

In this example, the transformation formula specifies the transformation mechanism for an incoming pipe separated delimited file. The formula's return values are the same as the list of attributes in the template file for the User Defined Table business object.

And this code snippet has the formula for this example.

```
/* Inputs */
INPUTS ARE OPERATION (text), LINENO (number), LINEREPEATNO (number),POSITION1 (text), POSITION2 (text),
 POSITION3 (text), POSITION4 (text), POSITION5 (text), POSITION6 (text), POSITION7 (text), POSITION8 (text)
DEFAULT FOR POSITION1 IS 'NO DATA'
DEFAULT FOR POSITION2 IS 'NO DATA'
DEFAULT FOR POSITION3 IS 'NO DATA'
DEFAULT FOR POSITION4 IS 'NO DATA'
```

```
DEFAULT FOR POSITION5 IS 'NO DATA'
DEFAULT FOR LINEREPEATNO IS 1
IF OPERATION='FILETYPE' THEN
OUTPUTVALUE='DELIMITED'
ELSE IF OPERATION='DELIMITER' THEN
OUTPUTVALUE='|'
ELSE IF OPERATION='READ' THEN
OUTPUTVALUE='NONE'
ELSE IF OPERATION = 'NUMBEROFBUSINESSOBJECTS' THEN
(
OUTPUTVALUE = '1'/*Always be 1*/
RETURN OUTPUTVALUE
)
ELSE IF OPERATION = 'METADATALINEINFORMATION' THEN
(
METADATA1[1] = 'SMUDT' /*TemplateCode*/
METADATA1[2] = 'UserDefinedTable' /*FileDiscriminator*/
METADATA1[3] = 'UserDefinedTable_UserTableCode'
METADATA1[4] = 'UserDefinedTable_LegislativeDataGroupName'
METADATA1[5] = 'UserDefinedTable_UserTableName'
METADATA1[6] = 'UserDefinedTable_RangeOrMatch'
METADATA1[7] = 'UserDefinedTable_UserRowTitle'
METADATA1[8]= 'UserDefinedTable_UserKeyUnits'
RETURN METADATA1 /*You can return only one METADATA for the respective template*/
)
ELSE IF OPERATION='MAP' THEN
(
FileName = 'SMUDT'
BusinessOperation = 'HSDL'
FileDiscriminator = 'UserDefinedTable'
UserDefinedTable_UserTableCode = POSITION1
UserDefinedTable_LegislativeDataGroupName = POSITION2
UserDefinedTable_UserTableName = POSITION1
UserDefinedTable_RangeOrMatch = POSITION3
UserDefinedTable_UserRowTitle = POSITION4
UserDefinedTable_UserKeyUnits = POSITION5
RETURN
 BusinessOperation,FileDiscriminator,FileName,UserDefinedTable_UserTableCode,UserDefinedTable_LegislativeDataGroupNa
)
ELSE
OUTPUTVALUE='NONE'
RETURN OUTPUTVALUE
/* End Formula Text */
```

Note: The template code needs to be used in the Fast Formula.

Related Topics

- Guidelines for Using HCM Spreadsheet Data Loader
- How Data Is Uploaded Using HCM Spreadsheet Data Loader
- HCM Spreadsheet Data Loader Templates
- Create and Edit Spreadsheet Templates
- Guidelines for Designing Spreadsheet Templates

21 Loading Payroll Interface Records

Overview of Loading Payroll Interface Inbound Records

Use the Payroll Interface Inbound Record object to import a third-party provider's processed payroll data into Oracle Human Capital Management Cloud.

Using HCM Data Loader, you can load these record types into the HCM cloud:

- Processed payroll data
- Generated payslips

You can use the imported payroll data to create custom reports in the OTBI application and to view imported payslips.

Payroll Interface Inbound Records Hierarchy

This table describes the two Payroll Interface Inbound Records components:

Component	What It Stores
Payroll Interface Inbound Record	Summary information from third-party payroll providers, such as the name and payroll period of the processed payroll.
Payroll Interface Inbound Record Information	Details about payrolls processed by third-party payroll providers, such as earnings, deductions, and messages.

Here are some of the considerations for loading Payroll Interface Inbound Records.

Extensible FlexField

You can use the Payroll Interface Inbound Record EFF extensible flexfield to load inbound data from third-party applications.

The HCM Data Loader application creates different contexts with predefined segments for the processed payroll values.

Record Type

You can specify the Record Type in the extensible flexfield. These two record types are supported - payroll-processed data and generated payslip files.

Loading Generated Payslips

Use the Document of Record object to load the payslips. You must load this data either with or before loading the payroll interface inbound record that references it.

Overview of Payroll Interface Inbound Record Component

The Payroll Interface Inbound Record component holds summary information from third-party payroll providers, such as the name and payroll period of the processed payroll. Use the `PayrollInterfaceInboundRecord` discriminator to load `PayrollInterfaceInboundRecord` records.

When loading a new inbound record, you supply these `PayrollInterfaceInboundRecord` attributes.

Attribute	What It's
`SourceSystemId / SourceSystemOwner`	A unique reference for the record being created. Supply either a source key or one of these user key attributes: `RecordType, PayrollName, LegislativeDataGroupName, PeriodName, BatchCode, FunctionalCategory, SourceType, and EntityIdentifier`.
`FunctionalCategory`	The functional category of the inbound record that the lookup `ORA_HRY_INBD_FUNCTIONAL_CATG` validates.
`RecordType`	The type of the inbound record that the lookup `ORA_HRY_INBD_REC_RECORD_TYPE` validates. The application supports these two record types: Payroll Data and Payslip Data.
`BatchCode`	The batch or file identifier.
`SourceType`	The type of entity. Valid source types include Person, Payroll Terms, Payroll Assignment, and Payroll Relationship. When supplying the payslip data, supply the Payroll Relationship.
`EntityIdentifier`	The number or name of entity.
`LegislativeDataGroupName`	The name of the legislative data group to store this record.
`PayrollName`	The name of the payroll.
`DocumentType`	The type of document used to store the third-party payslip. The value is required when the **RecordType** is Payslip Data and must be Third Party Payslip.
`DocumentCode`	The document code to uniquely identify the third-party payslip. This value is required when the `RecordType` is Payslip Data.
`StartDate`	The start date of the Inbound Record. The format is `yyyy/mm/dd hh24:mi:ss`.
`PeriodName`	The period to which the record relates.

Attribute	What It's
`CategoryCode`	The category code of the record. For Third Party Payslip data, specify `ORA_HRY_INBD_PAYSLIP`. When supplying data for other context, specify `ORA_HRY_INBD_PAYROLL`.
`BatchDate`	The batch or file date.

Overview of Payroll Interface Inbound Record Information Component

The Payroll Interface Inbound Record Information component holds the details of payroll data processed by third-party payroll providers, such as earning, deductions, and messages.

Use the `PayrollInterfaceInboundRecordInfo` discriminator to load Payroll Interface Inbound Record Information records.

When loading a new inbound information record, you supply these `PayrollInterfaceInboundRecordInfo` attributes.

Attribute	What It's
`SourceSystemId / SourceSystemOwner`	A unique reference for the record. Supply either a source key or one of these user key attributes: `RecordType`, `PayrollName`, `LegislativeDataGroupName`, `PeriodName`, `BatchCode`, `FunctionalCategory`, `SourceType`, `EntityIdentifier`, and `IriInformationContext`
`FLEX:ORA_HRY_PI_INBD_RECORDS_INFO_EFF`	The context code for the category of data being loaded.
`EFF_CATEGORY_CODE`	The code for the EFF Category. For third party payslip information, enter `ORA_HRY_INBD_PAYSLIP`. For other context, enter `ORA_HRY_INBD_PAYROLL`.
`FunctionalCategory`	The functional category of the inbound record. The `ORA_HRY_INBD_FUNCTIONAL_CATG` lookup validates this category. This value must match up with the parent record's functional category.
`RecordType`	The type of inbound record. This value must match up with the parent record's record type.
`BatchCode`	The batch or file identifier. This value must match up with parent record's batch code.
`SourceType`	The type of entity. Valid source types are Person, Payroll Terms, Payroll Assignment, and Payroll Relationship. This value must match up with the parent record's source type. When supplying the Payslip data, supply Payroll Relationship.
`EntityIdentifier`	The number or name of the entity. Enter the value depending upon the `SourceType`.
`LegislativeDataGroupName`	The name of the legislative data group. This value must match up with the parent record's legislative data group name.

Attribute	What It's
PayrollName	The record's payroll name. This value must match up with the parent record's payroll name.
StartDate	The start date of the Inbound Record. The format is yyyy-mm-dd. Note that this format is different from that of the parent record.
PeriodName	The period name of the record. This value must match up with the parent record's period name.
IriInformationContext	The context code for the category of data. This value is the same as for the FLEX:ORA_HRY_PI_INBD_RECORDS_INFO_EFF attribute.

When loading payment information, you must enter these extensible flexfield attributes:

Attribute	What It's
PaymentType(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)	The type of payment, such as cash, check, or electronic funds transfer.
CheckNumber(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)	The check number issued to the employee.
BankName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)	The employee's bank name to transfer electronic funds payments.
BranchName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)	The employee's branch name to transfer electronic funds payments.
AccountName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)	The employee's bank account name to transfer electronic funds payments.
AccountNumber(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)	The employee's bank account number to transfer electronic funds payments.
IBAN(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)	The international bank account number to facilitate the communication and processing of cross-border transactions.
Amount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)	The amount paid to the employee.
Currency(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)	The currency of the payment.

Attribute	What It's
PaymentDate(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)	The date the instruction was given to transfer the payment from the source bank account to the employee's bank account.
PayrollRunType(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)	The type of payroll run, such as regular or supplemental.
PayrollRunSequence(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)	The order in which payroll run type is processed.
ElementClassification(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)	The primary element classification, such as Standard Earnings or Supplemental Earnings.
ElementName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)	The name of the element.
CostCenter(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)	The cost center for employee payments.
CurrentAmount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)	The total amount for the element in the current payroll period.
QuarterToDateAmount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)	The total amount for the element in the current quarter.
YearToDateAmount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)	The total amount for the element in the current financial year.
Currency(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)	The currency of the payment.
GeneralLedgerCreditAccount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)	The code of the General Ledger account to credit earnings or deductions.
GeneralLedgerDebitAccount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)	The code of the General Ledger account to debit earnings or deductions.

When loading third-party payslip information, you must enter these extensible flexfield attributes:

Attribute	What It's
NetPayment(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_THIRD_PARTY_PAYSLIP)	The net payment amount.
PayslipViewDate(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_THIRD_PARTY_PAYSLIP)	

When loading absence information, you must enter these extensible flexfield attributes:

Attribute	What It's
AbsenceName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)	The name of the absence for the employee.
AbsenceBalance(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)	The balance of hours or days remaining for the balance.
AbsenceUnit(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)	The total number of absence units.
AbsenceUnitOfMeasure(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)	The type of unit of measure, such as day or hour.
AbsenceMessage(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)	The message from the third-party payroll provider containing absence information, such as remaining balance for the employee.

When loading message information, you must enter these extensible flexfield attributes.

Attribute	What It's
MessageType(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_MESSAGE_INFO)	The severity level of the message.
MessageId(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_MESSAGE_INFO)	A short code for the message received from the third-party payroll provider.

Attribute	What It's
MessageText(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_MESSAGE_INFO)	The message from the third-party payroll provider containing information about any errors that occurred during file upload.
OutboundFileName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_MESSAGE_INFO)	The name of the file sent to the third-party payroll provider.
OutboundFileReceivedDate(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_MESSAGE_INFO)	The date when the outbound file was received by the third-party payroll provider.

Examples for Loading Payroll Interface Inbound Records

The payroll interface inbound records dat file data is classified into the Inbound Record and Inbound Record Info sections.

Example for Payroll Interface Inbound Records

This example provides the parent records for the Payroll Interface Inbound Record object. Source keys aren't supplied, because this data is usually sourced from a third-party payroll provider with no knowledge of existing source key values.

Here's an example for the child Payroll Interface Inbound Record Information.

```
METADATA|PayrollInterfaceInboundRecord|SourceType|EntityIdentifier|FunctionalCategory|RecordType|PeriodName|
StartDate|BatchCode|BatchDate|VendorCode|LegislativeDataGroupName|PayrollName|CategoryCode
MERGE|PayrollInterfaceInboundRecord|Person|955160008166185|ADP Payforce|Payroll Data|1 2015 Monthly
 (Calendar)||Monthly Payroll Data Jan15|2015/1/31 00:00:00|ADP|Vision Corporation US LDG|ZHRX_US_PayrollOne|
ORA_HRY_INBD_PAYROLL
MERGE|PayrollInterfaceInboundRecord|Person|955160008166527|ADP Payforce|Payroll Data|1 2015 Monthly
 (Calendar)||Monthly Payroll Data Jan15|2015/1/31 00:00:00|ADP|Vision Corporation US LDG|ZHRX_US_PayrollOne|
ORA_HRY_INBD_PAYROLL
MERGE|PayrollInterfaceInboundRecord|Person|955160008166310|ADP Payforce|Payroll Data|1 2015 Monthly
 (Calendar)||Monthly Payroll Data Jan15|2015/1/31 00:00:00|ADP|Vision Corporation US LDG|ZHRX_US_PayrollOne|
ORA_HRY_INBD_PAYROLL
MERGE|PayrollInterfaceInboundRecord|Person|955160008166310|ADP Payforce|Payroll Data|2 2015 Monthly
 (Calendar)||Monthly Payroll Data Feb15|2015/1/31 00:00:00|ADP|Vision Corporation US LDG|ZHRX_US_PayrollOne|
ORA_HRY_INBD_PAYROLL
MERGE|PayrollInterfaceInboundRecord|Payroll Terms|ET955160008166310|ADP Payforce|Payroll Data|1 2015 Monthly
 (Calendar)||Monthly Payroll Data Jan15|2015/1/31 00:00:00|ADP|Vision Corporation US LDG|ZHRX_US_PayrollOne|
ORA_HRY_INBD_PAYROLL
MERGE|PayrollInterfaceInboundRecord|Payroll Assignment|E955160008166310|ADP Payforce|Payroll Data|1
 2015 Monthly (Calendar)||Monthly Payroll Data Jan15|2015/1/31 00:00:00|ADP|Vision Corporation US LDG|
ZHRX_US_PayrollOne|ORA_HRY_INBD_PAYROLL
```

Example for Payroll Interface Inbound Record Information

The Inbound Record Information component uses descriptive flexfields to support each record type. Each record type has its own flexfield context structure. You can provide these record types using the Payroll Interface Inbound Record object.

You can provide these record types using the Payroll Interface Inbound Record object.

- Payment Info
- Payroll Info
- Message Info
- Absence Info

As shown in these scenarios, you can either load all flexfield context information in a single file or load with individual flexfield context information.

Example 1: This .dat file has all flexfield context information for an employee.

```
METADATA|PayrollInterfaceInboundRecordInfo|SourceType|EntityIdentifier|RecordType|FunctionalCategory|
LegislativeDataGroupName|PayrollName|PeriodName|BatchCode|FLEX:ORA_HRY_PI_INBD_RECORDS_INFO_EFF|
IriInformationContext|EFF_CATEGORY_CODE|PaymentType(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)|
CheckNumber(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)|
BankName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)|
BranchName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)|
BranchNumber(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)|
AccountName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)|
AccountNumber(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)|
IBAN(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)|
Amount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)|
Currency(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)|
PaymentDate(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYMENT_INFO)|
PayrollRunType(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
PayrollRunSequence(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
ElementClassification(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
ElementName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
CurrentAmount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
QuarterToDateAmount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
YearToDateAmount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
Currency(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
CostCenter(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
GeneralLedgerDebitAccount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
GeneralLedgerCreditAccount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
MessageType(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_MESSAGE_INFO)|
MessageId(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_MESSAGE_INFO)|
MessageText(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_MESSAGE_INFO)|
OutboundFileName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_MESSAGE_INFO)|
OutboundFileReceivedDate(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_MESSAGE_INFO)|
AbsenceName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)|
AbsenceBalance(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)|
AbsenceUnit(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)|
AbsenceUnitOfMeasure(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)|
AbsenceMessage(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166527|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|1 2015 Monthly (Calendar)|Monthly Payroll Data Jan15|ORA_HRY_PAYMENT_INFO|
ORA_HRY_PAYMENT_INFO|ORA_HRY_INBD_PAYROLL|Direct Deposit1|12345|HDFC2|ABCD|D0231|Gerry Mars|90004|876454224|
13000|INR|2015/01/28|||||||||||||||||||||
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166527|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|1 2015 Monthly (Calendar)|Monthly Payroll Data Jan15|ORA_HRY_PAYROLL_INFO|
ORA_HRY_PAYROLL_INFO|ORA_HRY_INBD_PAYROLL||||||||||||General Payroll|1|EARNINGS|BASIC PAY|550|550|550|INR|
G1234|900-00-456|800-00-456|||||||||||
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166527|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|1 2015 Monthly (Calendar)|Monthly Payroll Data Jan15|ORA_HRY_PAYROLL_INFO|
ORA_HRY_PAYROLL_INFO|ORA_HRY_INBD_PAYROLL||||||||||||General Payroll|1|DEDUCTIONS|VOLUNTARY DEDUCTION|550|
550|550|INR|G1234|900-00-456|800-00-456|||||||||||
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166527|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|1 2015 Monthly (Calendar)|Monthly Payroll Data Jan15|ORA_HRY_PAYROLL_INFO|
ORA_HRY_PAYROLL_INFO|ORA_HRY_INBD_PAYROLL||||||||||||General Payroll|1|DEDUCTIONS|INCOME TAX|550|550|550|
INR|G1234|900-00-456|800-00-456|||||||||||
```

```
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166527|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|1 2015 Monthly (Calendar)|Monthly Payroll Data Jan15|ORA_HRY_MESSAGE_INFO|
ORA_HRY_MESSAGE_INFO|ORA_HRY_INBD_PAYROLL|||||||||||||||||||||||||ERROR|ERROR-211|Invalid Employee Number|
OUBNDJNJPAYROLL23|2014/01/24|||||||
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166527|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|1 2015 Monthly (Calendar)|Monthly Payroll Data Jan15|ORA_HRY_LEAVE_INFO|
ORA_HRY_LEAVE_INFO|ORA_HRY_INBD_PAYROLL|||||||||||||||||||||||||||||CASUAL|8|4|Day|||
```

Scenario 2: This example shows how to load only the payroll info flexfield context.

```
METADATA|PayrollInterfaceInboundRecordInfo|SourceType|EntityIdentifier|RecordType|
FunctionalCategory|LegislativeDataGroupName|PayrollName|PeriodName|BatchCode|
FLEX:ORA_HRY_PI_INBD_RECORDS_INFO_EFF|IriInformationContext|EFF_CATEGORY_CODE|
ElementClassification(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
ElementName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
CurrentAmount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
QuarterToDateAmount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
YearToDateAmount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
Currency(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
CostCenter(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
GeneralLedgerDebitAccount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
GeneralLedgerCreditAccount(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
PayrollRunSequence(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)|
PayrollRunType(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_PAYROLL_INFO)
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166527|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|1 2015 Monthly (Calendar)|Monthly Payroll Data Jan15|ORA_HRY_PAYROLL_INFO|
ORA_HRY_PAYROLL_INFO|ORA_HRY_INBD_PAYROLL|EARNINGS|BASIC PAY|10000|10000|10000|INR|G1234|900-00-456|
800-00-456||
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166527|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|1 2015 Monthly (Calendar)|Monthly Payroll Data Jan15|ORA_HRY_PAYROLL_INFO|
ORA_HRY_PAYROLL_INFO|ORA_HRY_INBD_PAYROLL|DEDUCTIONS|INCOME TAX|550|550|550|INR|G1234|900-00-456|
800-00-456||
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166527|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|1 2015 Monthly (Calendar)|Monthly Payroll Data Jan15|ORA_HRY_PAYROLL_INFO|
ORA_HRY_PAYROLL_INFO|ORA_HRY_INBD_PAYROLL|DEDUCTIONS|VOLUNTARY DEDUCTION|800|800|800|INR|G1234|900-00-456|
800-00-456||
```

Scenario 3: This example shows how to load only the Absence Info flexfield context.

```
METADATA|PayrollInterfaceInboundRecordInfo|SourceType|EntityIdentifier|RecordType|FunctionalCategory|
LegislativeDataGroupName|PayrollName|PeriodName|BatchCode|FLEX:ORA_HRY_PI_INBD_RECORDS_INFO_EFF|
IriInformationContext|EFF_CATEGORY_CODE|AbsenceName(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)|
AbsenceBalance(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)|
AbsenceUnit(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)|
AbsenceMessage(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)|
AbsenceUnitOfMeasure(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_LEAVE_INFO)
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166527|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|1 2015 Monthly (Calendar)|Monthly Payroll Data Jan15|ORA_HRY_LEAVE_INFO|
ORA_HRY_LEAVE_INFO|ORA_HRY_INBD_PAYROLL|CASUAL|8|4||
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166310|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|1 2015 Monthly (Calendar)|Monthly Payroll Data Jan15|ORA_HRY_LEAVE_INFO|
ORA_HRY_LEAVE_INFO|ORA_HRY_INBD_PAYROLL|VACATION|20|4||
MERGE|PayrollInterfaceInboundRecordInfo|Person|955160008166310|Payroll Data|ADP Payforce|Vision Corporation
 US LDG|Monthly Payroll|2 2015 Monthly (Calendar)|Monthly Payroll Data Feb15|ORA_HRY_LEAVE_INFO|
ORA_HRY_LEAVE_INFO|ORA_HRY_INBD_PAYROLL|VACATION|20|8||
```

Example of Importing Generated Payslips

Use HCM Data Loader to load a payslip that was generated in compressed format, from a third-party application into Oracle HCM Cloud.

The application uses the `PayrollInterfaceInboundRecord` discriminator to load already generated payslips with the same attributes that are supplied when loading employee's payroll data.

The application uses the `PayrollInterfaceInboundRecord` discriminator to load the already generated payslips with the same attributes that are supplied when loading employee's payroll data.

The application uses the `PayrollInterfaceInboundRecord` discriminator to load the already generated payslips with the same attributes that are supplied when loading employee's payroll data.

> **Note:** The application uses the `PayrollInterfaceInboundRecord` discriminator to load already generated payslips with the same attributes that are supplied when loading employee's payroll data.

```
METADATA|PayrollInterfaceInboundRecord|InbdRecordId|SourceType|EntityIdentifier|FunctionalCategory|
RecordType|PeriodName|StartDate|BatchCode|BatchDate|VendorCode|LegislativeDataGroupName|PayrollName|
CategoryCode|DocumentCode|DocumentType
MERGE|PayrollInterfaceInboundRecord||Payroll Relationship|955160008173009|ADP Payforce|Payslip Data|11
 2015 Monthly (Calendar)||Monthly Payroll Data Nov15|2015/11/01 00:00:00|ADP|Vision Corporation US LDG|
ZHRX_US_PayrollOne|ORA_HRY_INBD_PAYSLIP|ThirdPartyPayslip_955160008173009_Nov2015|Third Party Payslip
METADATA|PayrollInterfaceInboundRecordInfo|FLEX:ORA_HRY_PI_INBD_RECORDS_INFO_EFF|EFF_CATEGORY_CODE|
InbdRecordInfoId|InbdRecordId|SourceType|RecordType|BatchCode|FunctionalCategory|EntityIdentifier|
LegislativeDataGroupName|PayrollName|PeriodName|IriInformationContext|SourceSystemOwner|
SourceSystemId|GUID|NetPayment(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_THIRD_PARTY_PAYSLIP)|
PayslipViewDate(ORA_HRY_PI_INBD_RECORDS_INFO_EFF=ORA_HRY_THIRD_PARTY_PAYSLIP)
MERGE|PayrollInterfaceInboundRecordInfo|ORA_HRY_THIRD_PARTY_PAYSLIP|ORA_HRY_INBD_PAYSLIP|||Payroll
 Relationship|Payslip Data|Monthly Payroll Data Nov15|ADP Payforce|955160008173009|Vision Corporation
 US LDG|ZHRX_US_PayrollOne|11 2015 Monthly (Calendar)|ORA_HRY_THIRD_PARTY_PAYSLIP|HRC_SQLLOADER|
ADP_955160008173009_ORA_HRY_THIRD_PARTY_PAYSLIP_0101||5550|2015/11/20
```

The HCM Data Loader application uploads the payslips that are generated by a third-party application into the Document of Record business object.

The payslip data has two sections.

1. The first part is a data file that contains a list of employees for that require payslips and their respective payslip file names. You must name the payslip data file as `DocumentsOfRecord.dat`.
2. The second part is the collection of actual payslip files. You must provide this dat file in the form of a compressed file and group these files into a `BlobFiles` subfolder. This file must contain the `DocumentsOfRecord.dat` file and the subfolder BlobFiles containing the payslip files. You use this compressed file for loading the data.

In this `DocumentsOfRecord.dat` file, the `DocumentsOfRecord` discriminator is used for the header record and the `DocumentAttachment` record provides the attachment details. You must reference the compressed file placed in the BlobFiles folder by name against the File attribute in the `DocumentAttachment` record.

```
METADATA|DocumentsOfRecord|PersonNumber|DocumentType|Country|DocumentCode|DocumentName|DocumentNumber|
DateFrom|DateTo|IssuingAuthority|IssuedDate|IssuingCountry|IssuingLocation|Comments|Publish|PublishDate|
Status|VerifiedBy|VerifiedDate
MERGE|DocumentsOfRecord|955160008173009|Third Party Payslip||ThirdPartyPayslip_955160008173009_Nov2015|
USThirdPartyPayslip||2015/11/01|2015/11/30||2015/11/30||||||||
METADATA|DocumentAttachment|PersonNumber|DocumentType|Country|DocumentCode|DataTypeCode|Title|
URLorTextorFileName|File
MERGE|DocumentAttachment|955160008173009|Third Party Payslip||ThirdPartyPayslip_955160008173009_Nov2015|
FILE|November Payslip|USOnlinePayslip_November_2015.pdf|USOnlinePayslip_November_2015.pdf
```

22 Loading Payroll Localization Data for Canada

Geography Codes for Canadian Provinces

Province codes are used when initializing balances and loading province level calculation card information.

Balances

The value entered in the Area One contexts in the batch lines should be the numerical geography code for the province. For example, '19' represents Quebec. The full list of geography codes are listed below for each province.

Calculation Cards

The Context1 attribute on the card component needs to be supplied for the following calculation cards:

- Calculation Rules for Tax Reporting and Payroll Statutory Unit
- Tax Credit Information

Geography Codes

Province	Geocode Value
Alberta	1
British Columbia	3
Manitoba	5
New Brunswick	7
Newfoundland and Labrador	9
Northwest Territories	11
Nova Scotia	13
Ontario	15
Prince Edward Island	17
Quebec	19
Saskatchewan	21
Yukon	23
Nunavut	25

Payroll Assignment Information for Canada

Using the Payroll Assignment Details object, you can create or update payroll assignment details for an employee, such as time card required status and overtime period. However, you can't delete these details.

You can load assignment level Canadian information, using the HCM Data Loader (HDL) for the **Payroll Assignment Details** object. You can add the following additional information for Canada:

- Indian Exempt Status
- Workers Compensation Exempt
- Workers Compensation Override Information, such as:
    - Workers Compensation Province Override
    - Workers Compensation Account Override
    - Workers Compensation Classification Unit Override
- Provincial Medical Exempt
- Provincial Medical Override Information, such as:
    - Provincial Medical Province Override
    - Provincial Medical Account Override
- Vacation Liability Province Override

Involuntary Deduction Information

Overview of Involuntary Deduction Cards for Canada

An involuntary deduction is a court-ordered payment taken from the employee's pay and then paid to a court or an individual person.

The deduction details for each involuntary deduction are captured on an Involuntary Deduction calculation card, which is attached to the employee.

Considerations and Prerequisites

A number of prerequisites must be configured before loading the Involuntary Deductions card or the card components:

- Involuntary deduction elements
- Element eligibility
- Third-party payees
- Third-party payment methods

Create a card component for each involuntary deduction that an employee has, with the appropriate values provided. The enterable calculation values can vary by involuntary deduction type. Refer to the Cloud Customer Connect topic *BI Publisher Report: Canada - Involuntary Deductions Value Definitions* for a report that extracts the value definitions for the Involuntary Deduction elements you have defined.

You can find more information about Involuntary Deduction cards for Canada, including how to create the prerequisite data structures in the Canada: Payroll Involuntary Deductions technical brief. Refer to the Canada Information Center on My Oracle Support:

https://support.oracle.com/rs?type=doc&id=2102586.2

Navigation: CA –Payroll tab > Technical Briefs: Involuntary Deductions

Involuntary Deductions Card Record Types

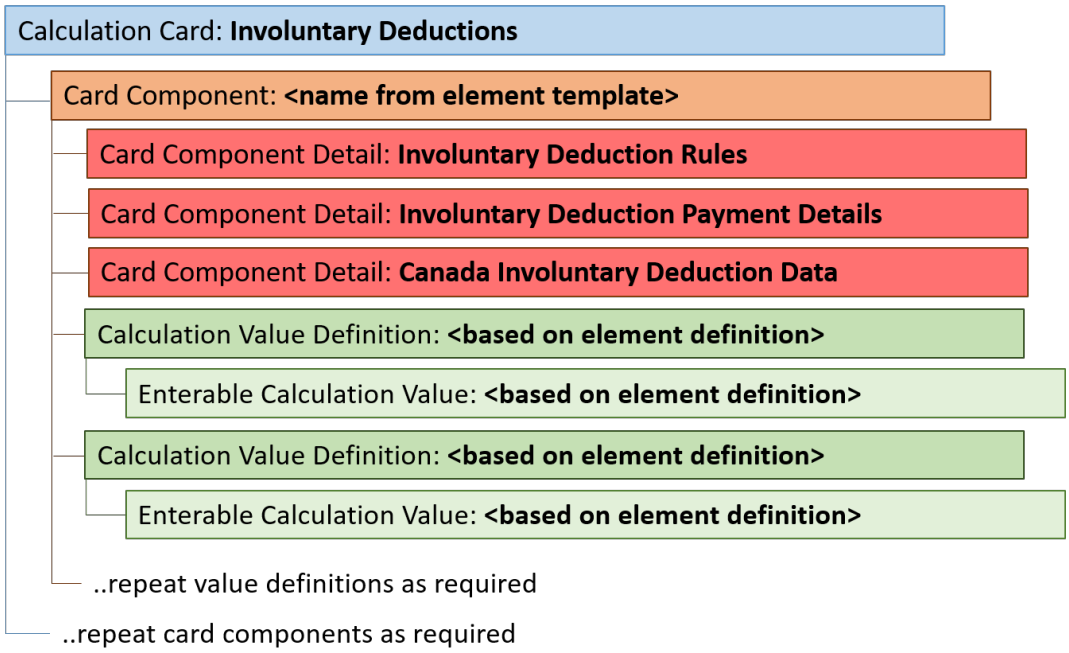
The Involuntary Deduction card is bulk-loaded using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various localization requirements.

The Involuntary Deduction Card uses the following Calculation Card record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee payroll relationship that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following topics describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Supply a component detail record for each flexfield context required by each card component.	ComponentDetail
Calculation Value Definition	Allows the creation of value definitions so that overriding values can be specified on the card component. Details of the specific value definitions are provided in the following sections.	CalculationValueDefinition
Enterable Calculation Value	Used to specify an overriding value for each calculation value definition.	EnterableCalculationValue

Involuntary Deduction Calculation Card Hierarchy

The hierarchy of Calculation Card components applicable to Involuntary Deductions is dependent upon the type of involuntary deduction being reported, but generally has this shape:



A separate card component is required for each involuntary deduction type. There is only one Involuntary Deduction card for an employee.

The card component name is defined by the element configuration, as are the value definitions, which are supplied using the **Calculation Value Definition** and **Enterable Calculation Value** record types.

There are three flexfield contexts available, of which the **Canada Involuntary Deduction Data** context is optional. Data for these is loaded using the Component Detail record type.

Refer to the topic **Loading Involuntary Deductions Cards for Canada** for details of the attributes to supply.

Related Topics

- Guidelines for Loading Calculation Cards

Load Involuntary Deduction Cards for Canada

A Calculation Card record needs to be created for every employee for whom you are maintaining involuntary deductions.

Even if you are updating an existing calculation card and the calculation card itself is not being updated, still include the calculation card record to group other related data supplied in the file.

Involuntary Deductions Calculation Card Attributes

The Involuntary Deductions calculation card for employees uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Involuntary Deductions calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Involuntary Deductions'
CardSeqeunce	N/A	Specify '1'. Not required when source keys are used.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card. Format YYYY/MM/DD.

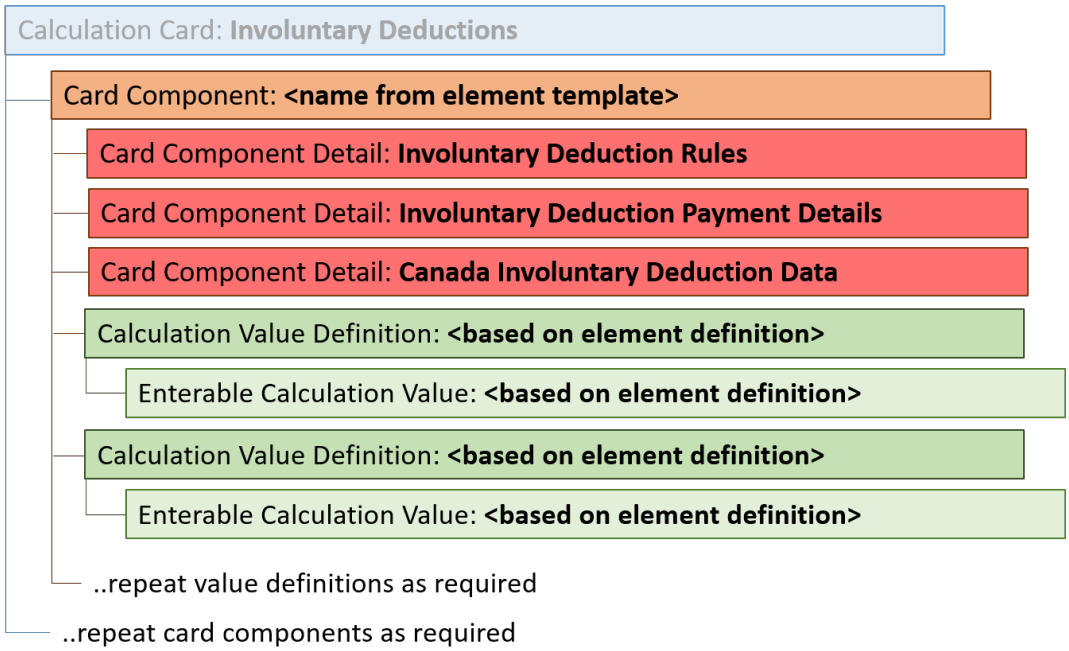
These attributes are supplied against the CalculationCard file discriminator.

Loading a Card Component for Involuntary Deductions

Each Involuntary Deduction element is created as an individual card component on the same Involuntary Deduction card for each employee. An employee will only have one Involuntary Deduction card.

This topic provides the general guidance on how to create a Card Component for an Involuntary Deduction. Also review the topic for the specific type of involuntary deduction you are loading.

The general card component hierarchy for an Involuntary Deduction:



The card component name is defined by the element configuration, as are the value definitions which are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Use the Canadian Involuntary Deductions BI Publisher report to extract the value definitions applicable for the element. Refer to the Cloud Customer Connect topic: *BI Publisher Report: Canada - Involuntary Deductions Value Definitions* for this report.

Card Component Attributes for Involuntary Deductions

All Involuntary Deductions card components uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Involuntary Deduction card component. For new card components, supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Involuntary Deduction calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The start date of the Involuntary Deduction card component.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify the name of the element defined for the involuntary deduction. Element names can also be found in the list of values for the Element parameter on the Canada Involuntary Deduction Value Definitions report.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Specify '1' if using user keys. Not required when source keys are used.
Context1	N/A	A reference code that uniquely identifies the order. An employee can have more than one order of the same deduction element. You can enter a court order number, remittance identifier, or other identifier provided by the issuing authority.

These attributes are supplied against the CardComponent file discriminator and must be supplied along with a CalculationCard record for the parent Involuntary Deductions card.

Component Detail Attributes for Involuntary Deductions

The component detail record type lets you to upload data into flexfield segments.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context and card component:

Card Component	Flexfield Context Code
Involuntary Deduction Rules	INVLN_DEDN_SUPPORT_DATA
Involuntary Deduction Payment Details	INVLN_DEDN_PAYEE_DETAILS
Canada Involuntary Deduction Data	ORA_HRX_CA_INV_DEDN_DATA

Core attributes for Component Details:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName,	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component
DirInformationCategory	N/A	The code for the flexfield context applicable to the card component. These are listed above this table. For example, supply 'INVLN_DEDN_SUPPORT_DATA' for the Involuntary Deduction Rules card component.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parent card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.

The attributes used to populate the flexfield segments differ depending on the flexfield context.

You can find the current flexfield segment attribute names for each flexfield context using the View Business Objects task.

> **Note:** Flexfield segments validated by a value set with both an ID and meaning value offer two HCM Data Loader attributes, the attribute ending _Display accepts the meaning. It is recommended that you use the base attribute and supply the ID or lookup code value. HCM Data Loader uses your environments base language when uploading data so there will be translation issues if you supply the_Display attribute using a different language.

Refer to the Cloud Customer Connect topic *BI Publisher Report:Third Party Payees for Involuntary Deduction Cards* for a report that extracts your third-party payees and their surrogate IDs, which can be supplied to the Order Amount Payee flexfield segment.

Calculation Value Definition Attributes

Involuntary Deductions card components use value definitions to supply values or override values.

The value definitions to supply are determined by the element used for the card component. Use the Canada Involuntary Deductions Value Definitions BI publisher report to extract the list of value definitions applicable to your element. Refer to Cloud Customer Connect topic: *BI Publisher Report: Canada - Involuntary Deductions Value Definitions*.

The Calculation Value Definition record type specifies the name of the value definition that you are supplying an override value for.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records, supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent involuntary deduction card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent involuntary deduction card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent involuntary deduction card component.
ValueDefinitionName	N/A	The name of the value being overridden. For example, for a Garnishment, this can be "Court Order Amount".
ValueType	N/A	The optional type of the value definition, such as Rate or Total amount. Supply a code from the lookup type PAY_DED_OVERRIDE_TYPE.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent involuntary deduction card component and a CalculationCard record for the owning Involuntary Deductions card.

Enterable Calculation Value Attributes

The Enterable Calculation Value provides the value for the value definition. It references the Calculation Value Definition record, which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records, supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes.
Value1	N/A	The value for the value definition identified by the parent calculation value definition record. You can enter an amount, percent, or a Boolean value. **Note:** If defining a percent, enter the value as a decimal. For example, enter '.50' for 50%. For the Boolean value, enter a 'Y' for yes, and a 'N' for No.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Example of Creating an Involuntary Deductions Card for Child Support for Canada

This example CalculationCard.dat file creates an Involuntary Deductions card for a Canadian employee to pay child support.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|E8192419_ID|CA LDG|Involuntary Deductions|2020/01/01|E8192419

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|E8192419_ID_CS|CA LDG|E8192419|2020/01/01|SD MAINT 41219|35233

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|FLEX:Deduction
 Developer DF|orderAmtPayee_Display(Deduction Developer DF=INVLN_DEDN_PAYEE_DETAILS)|receivedDate(Deduction
 Developer DF=INVLN_DEDN_SUPPORT_DATA)
MERGE|ComponentDetail|VISION|E8192419_ID_CS_PD|CA LDG|E8192419_ID_CS|2020/01/01|SD MAINT 41219|
INVLN_DEDN_PAYEE_DETAILS|INVLN_DEDN_PAYEE_DETAILS|Ontario Courts|
MERGE|ComponentDetail|VISION|E8192419_ID_CS_DR|CA LDG|E8192419_ID_CS|2020/01/01|SD MAINT 41219|
INVLN_DEDN_SUPPORT_DATA|INVLN_DEDN_SUPPORT_DATA||2020/01/01

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName|ValueType
MERGE|CalculationValueDefinition|VISION|E8192419_ID_CS_CSOA|CA LDG|E8192419_ID_CS|2020/01/01|SD MAINT 41219|
Child Support Order Amount|T
MERGE|CalculationValueDefinition|VISION|E8192419_ID_CS_IF|CA LDG|E8192419_ID_CS|2020/01/01|SD MAINT 41219|
Child Support Initial Fee|T

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName|ValueType|Value1
MERGE|EnterableCalculationValue|VISION|E8192419_ID_CS_CSOA|CA LDG|E8192419_ID_CS_CSOA|2020/01/01|SD MAINT
 41219|Child Support Order Amount|T|350
MERGE|EnterableCalculationValue|VISION|E8192419_ID_CS_IF|CA LDG|E8192419_ID_CS_IF|2020/01/01|SD MAINT 41219|
Child Support Initial Fee|T|300
```

Note: This example uses the display attribute for the Order Amount Payee to supply the third-party payee name: orderAmtPayee_Display(Deduction Developer DF=INVLN_DEDN_PAYEE_DETAILS)

Refer to the Cloud Customer Connect topic: *BI Publisher Report: Third Party Payees for Involuntary Deduction Cards* for a report that extracts your third-party payees and their surrogate ID values. This can be used to populate the base flexfield attribute orderAmtPayee (Deduction Developer DF=INVLN_DEDN_PAYEE_DETAILS).

Example of Adding a Card Component to an Existing Canadian Involuntary Deduction Card

In this example, the source keys aren't known for the existing Involuntary Deductions card and so the card is referenced by user keys. The card component and its child records are identified by source keys.

This CalculationCard.dat file creates a Garnishment card component providing a Court Order Amount value definition override.

Refer to the Cloud Customer Connect topic: *BI Publisher Report: Third Party Payees for Involuntary Deduction Cards* for a report that extracts your third-party payees along with their surrogate ID values. This can be used to populate the flexfield attribute orderAmtPayee (Deduction Developer DF=INVLN_DEDN_PAYEE_DETAILS).

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|CardSequence|AssignmentNumber
MERGE|CalculationCard|||CA LDG|Involuntary Deductions|2018/01/01|1|E8192890

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|CardSequence|AssignmentNumber|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|E8175303_ID_GARN|CA LDG|Involuntary Deductions|2018/01/01|1|E8192890|Garnishment|
124555

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|FLEX:Deduction
 Developer DF|orderAmtPayee(Deduction Developer DF=INVLN_DEDN_PAYEE_DETAILS)|receivedDate(Deduction
 Developer DF=INVLN_DEDN_SUPPORT_DATA)
MERGE|ComponentDetail|VISION|E8175303_ID_GARN_PD|CA LDG|E8175303_ID_GARN|2018/01/01|Garnishment|
INVLN_DEDN_PAYEE_DETAILS|INVLN_DEDN_PAYEE_DETAILS|300100068244093|
MERGE|ComponentDetail|VISION|E8175303_ID_GARN_DR|CA LDG|E8175303_ID_GARN|2018/01/01|Garnishment|
INVLN_DEDN_SUPPORT_DATA|INVLN_DEDN_SUPPORT_DATA||2018/01/01

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|E8175303_ID_GARN_COA|CA LDG|E8175303_ID_GARN|2018/01/01|Garnishment|
Court Order Amount

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|VISION|E8175303_ID_GARN_COA|CA LDG|E8175303_ID_GARN_COA|2018/01/01|
Garnishment|Court Order Amount|120
```

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Flexfield Data

Organization Cards

Overview of Organization Calculation Cards for Canada

Organization calculation cards store values about federal and regional tax rules required for Oracle HCM processes.

Data entered on the calculation card can be either at the PSU or TRU calculation card level. When entered at the PSU level, the setup applies to all TRUs attached to the PSU. When entered at the TRU level, this information overrides the settings at the PSU level.

Considerations and Prerequisites

You must previously have the organizational structures for your business as required for HR and Payroll country-specific processes, defined. For further information, please refer to the Implementing Payroll for Canada on My Oracle Support at the location below:

Canada Information Center

https://support.oracle.com/rs?type=doc&id=2102586.2

Canada tab > Product Documentation > Payroll Guides > Implementing Payroll for Canada

Extract the Surrogate ID of the Tax Reporting Unit

The component sequence for the organization calculation card component is the surrogate ID of the tax reporting unit. It is an application-generated value to uniquely identify the tax reporting unit. You can use the Tax Reporting Unit BI Publisher report to extract your tax reporting units and their surrogate ID values. For details on how to import and run this report, refer to the Cloud Customer Connect topic: *BI Publisher Report: Tax Reporting Units by Legislation*.

> **Note:** The value can be different on different environments. For example, the value on the production pod can be different from the one on the test pod.

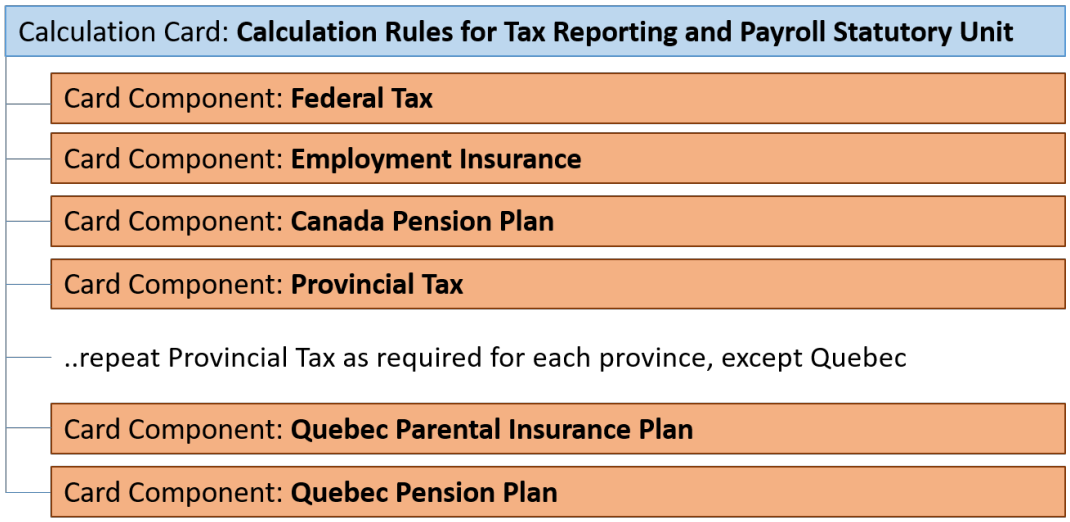
Calculation Card Record Types

The **Calculation Rules for Tax Reporting and Payroll Statutory Unit** calculation card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Calculation Rules for Tax Reporting and Payroll Statutory Unit card uses the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the payroll statutory unit that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Supply a component detail record for each flexfield context required by each card component.	ComponentDetail
Calculation Value Definition	Allows the creation of value definitions so that overriding values can be specified on the card component. Details of the specific value definitions are provided in the following sections.	CalculationValueDefinition
Enterable Calculation Value	For each calculation value definition, an overriding value can be specified using the Enterable Calculation Value record type.	EnterableCalculationValue

Calculation Rules for Tax Reporting and Payroll Statutory Unit Hierarchy



The Provincial Tax card component is repeated for each province, except Quebec, which uses the Quebec Parental Insurance Plan and Quebec Pension Plan card components instead.

The details of how to load data for each of these card components are found in the other related topics.

Refer to the topic: *Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards* for an example of creating a Calculation Rules for Tax Reporting and Payroll Statutory Unit card for a PSU.

Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards

Create one Calculation Rules for Tax Reporting and Payroll Statutory Unit card for each PSU and/or TRU that you have defined.

Even if you are updating an existing calculation card and the calculation card itself is not being updated, still include the calculation card record to group other related data supplied in the file.

Calculation Card Attributes

The Calculation Rules for Tax Reporting and Payroll Statutory Unit calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, DirCardDefinitionName, LegislativeDataGroupName And, PayrollStatutoryUnitName Or,	A unique identifier for the **Calculation Rules for Tax Reporting and Payroll Statutory Unit** calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	TaxRepUnitName	
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Calculation Rules for Tax Reporting and Payroll Statutory Unit'.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist. Not required when source keys are used.
PayrollStatutoryUnitName	N/A	The name of the Payroll Statutory Unit if the card is for a PSU. Don't supply for TRU level cards.
TaxRepUnitName	N/A	The name of the Tax Reporting Unit if the card is for a TRU. Don't supply for PSU level cards.
EffectiveStartDate	N/A	The start date of the calculation card.

These attributes are supplied against the CalculationCard file discriminator.

Supply the Calculation Card record along with the relevant card components as described in the following sections.

Common Card Component Attributes

Card components commonly use these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName And, PayrollStatutoryUnitName Or, TaxRepUnitName	A unique identifier for the card component. For new card components, supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, DirCardDefinitionName, LegislativeDataGroupName	The parent **Calculation Rules for Tax Reporting and Payroll Statutory Unit** calculation card should be identified by

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	And, PayrollStatutoryUnitName Or, TaxRepUnitName	using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card component. This must be equal to or after the EffectiveStartDate on the **Calculation Rules for Tax Reporting and Payroll Statutory Unit** calculation card. If updating an existing card component, the effective start date must be original start date of card the component.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Review the guidance for the card component being loaded for the value to supply.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.

These attributes are supplied against the CardComponent file discriminator.

Common Component Detail Attributes

The component detail record type allows you to upload data into flexfield segments.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield contexts used by the card component. Refer to the guidance for the card component being loaded.

You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Core attributes for Component Details:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, DirCardDefinitionName, DirCardCompDefName And, PayrollStatutoryUnitName Or, TaxRepUnitName	A unique identifier for the component detail record. For new component detail records, supply the source key attributes. You can also identify card components with the user key attributes.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName And, PayrollStatutoryUnitName Or, TaxRepUnitName	The parent card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parentcard component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component. Supply the same value as supplied to this attribute on the parent card component.
DirInformationCategory	N/A	The code for the flexfield context. Refer to the guidance for the card component for valid values.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Note: When supplying values to look up validated flexfield segments, you can provide the lookup meaning using the '_Display' suffixed attribute name, but it is recommended that you supply the lookup code to remove potential translation issues.

Example of Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit for Canada

This example calculation card creates the calculation rules for the 'CA PSU' payroll statutory unit:

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
CardSequence|EffectiveStartDate|PayrollStatutoryUnitName
MERGE|CalculationCard|VISION|LDGMAINCARD1|CA LDG|Calculation Rules for Tax Reporting and Payroll Statutory
 Unit|1|2002/01/01|CA PSU
```

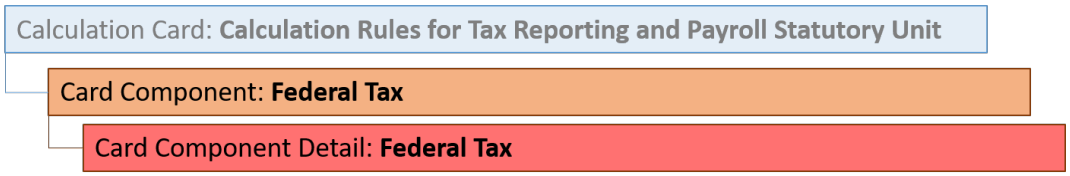
```
METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardCompDefName|
DirCardId(SourceSystemId)|EffectiveStartDate|ComponentSequence|Context1
MERGE|CardComponent|VISION|LDGMAINCARDCOMP1|CA LDG|Federal Tax|LDGMAINCARD1|2002/01/01|1|
MERGE|CardComponent|VISION|LDGMAINCARDCOMP2|CA LDG|Canada Pension Plan|LDGMAINCARD1|2002/01/01|1|
MERGE|CardComponent|VISION|LDGMAINCARDCOMP3|CA LDG|Employment Insurance|LDGMAINCARD1|2002/01/01|1|
MERGE|CardComponent|VISION|LDGMAINCARDCOMP4|CA LDG|Provincial Tax|LDGMAINCARD1|2002/01/01|19|19
MERGE|CardComponent|VISION|LDGMAINCARDCOMP5|CA LDG|Quebec Pension Plan|LDGMAINCARD1|2002/01/01|19|19
MERGE|CardComponent|VISION|LDGMAINCARDCOMP6|CA LDG|Quebec Parental Insurance Plan|LDGMAINCARD1|2002/01/01|
19|19

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardCompDefName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|ComponentSequence|DirInformationCategory|FLEX:Deduction
 Developer DF|_FEDERAL_NON_PERIODIC_TAX_METHO(Deduction Developer DF=HRX_CA_ORG_FEDERAL_INCOME_TAX)|
_FEDERAL_REGULAR_TAX_METHOD(Deduction Developer DF=HRX_CA_ORG_FEDERAL_INCOME_TAX)|
_CPP_SELF_ADJUST_METHOD(Deduction Developer DF=HRX_CA_ORG_FEDERAL_PENSION_PLAN)|
_EI_SELF_ADJUST_METHOD(Deduction Developer DF=HRX_CA_ORG_FEDERAL_EMPLOYMENT_INSURANCE)|
_PROVINCE_NON_PERIOD_TAX_METHOD(Deduction Developer DF=HRX_CA_ORG_PROVINCE_INCOME_TAX)|
_PROVINCE_REGULAR_TAX_METHOD(Deduction Developer DF=HRX_CA_ORG_PROVINCE_INCOME_TAX)|
_PPP_SELF_ADJUST_METHOD(Deduction Developer DF=HRX_CA_ORG_PROVINCE_PENSION_PLAN)|
_PPIP_SELF_ADJUST_METHOD(Deduction Developer DF=HRX_CA_ORG_PROVINCE_PARENTAL_INSURANCE_PLAN)
MERGE|ComponentDetail|VISION|AVCAVSCOMPDET2|CA LDG|Federal Tax|LDGMAINCARDCOMP1|2002/01/01|1|
HRX_CA_ORG_FEDERAL_INCOME_TAX|HRX_CA_ORG_FEDERAL_INCOME_TAX|FEDERAL_YTD_BONUS|FEDERAL_OPTION1||||||
MERGE|ComponentDetail|VISION|AVCAVSCOMPDET3|CA LDG|Canada Pension Plan|LDGMAINCARDCOMP2|2002/01/01|1|
HRX_CA_ORG_FEDERAL_PENSION_PLAN|HRX_CA_ORG_FEDERAL_PENSION_PLAN|||SELF_ADJ_AT_MAX|||||
MERGE|ComponentDetail|VISION|AVCAVSCOMPDET5|CA LDG|Employment Insurance|LDGMAINCARDCOMP3|2002/01/01|1|
HRX_CA_ORG_FEDERAL_EMPLOYMENT_INSURANCE|HRX_CA_ORG_FEDERAL_EMPLOYMENT_INSURANCE||||SELF_ADJ_AT_MAX||||
MERGE|ComponentDetail|VISION|AVCAVSCOMPDET1|CA LDG|Provincial Tax|LDGMAINCARDCOMP4|2002/01/01|
19|HRX_CA_ORG_PROVINCE_INCOME_TAX|HRX_CA_ORG_PROVINCE_INCOME_TAX|||||PROVINCE_BONUS_METHOD1|
PROVINCE_REGULAR_PAYMENTS||
MERGE|ComponentDetail|VISION|AVCAVSCOMPDET4|CA LDG|Quebec Pension Plan|LDGMAINCARDCOMP5|2002/01/01|19|
HRX_CA_ORG_PROVINCE_PENSION_PLAN|HRX_CA_ORG_PROVINCE_PENSION_PLAN|||||||SELF_ADJ_AT_MAX|
MERGE|ComponentDetail|VISION|AVCAVSCOMPDET6|CA LDG|Quebec Parental Insurance
 Plan|LDGMAINCARDCOMP6|2002/01/01|19|HRX_CA_ORG_PROVINCE_PARENTAL_INSURANCE_PLAN|
HRX_CA_ORG_PROVINCE_PARENTAL_INSURANCE_PLAN|||||||SELF_ADJ_AT_MAX


METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardDefinitionName|SourceId(SourceSystemId)|EffectiveStartDate|PayrollStatutoryUnitName|
TaxReportingUnitName|ComponentSequence|CardSequence|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|AVCAVSVD1|CA LDG|Calculation Rules for Tax Reporting and Payroll
 Statutory Unit|LDGMAINCARDCOMP3|2002/01/01|CA PSU||1|1|Employment Insurance|Employer EI Rate

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardDefinitionName|ValueDefnId(SourceSystemId)|EffectiveStartDate|PayrollStatutoryUnitName|
TaxReportingUnitName|ComponentSequence|CardSequence|DirCardCompDefName|ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|VISION|AVCAVSRI1|CA LDG|Calculation Rules for Tax Reporting and Payroll
 Statutory Unit|AVCAVSVD1|2002/01/01|CA PSU||1|1|Employment Insurance|Employer EI Rate|10
```

Guidelines for Loading Federal Tax Card Components for Canadian Organization Cards

The Federal Tax card component is used to capture information that will impact Federal Tax calculations for employees.

Federal Tax Card Component Hierarchy

The Federal Tax card component has this shape:



The Federal Tax card component uses the Federal Tax flexfield context and data for this is loaded using the Component Detail record type.

Refer to the *Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards* topic for the attributes to supply for each of these record types.

Card Component Name

When defining the Federal Tax card component, specify a value of 'Federal Tax' for the DirCardCompDefName attribute on the Card Component and Component Detail records.

Component Detail Flexfield Context

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context:

- Federal Tax (HRX_CA_ORG_FEDERAL_INCOME_TAX)

Note: You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Related Topics

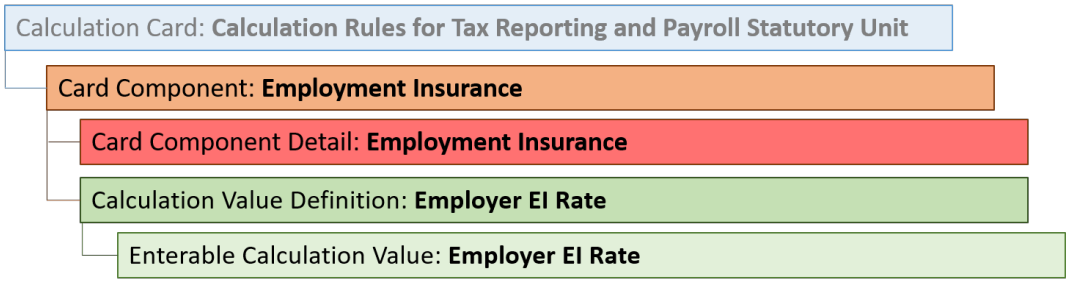
- Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards

Guidelines for Loading Employment Insurance Card Components for Canadian Organization Cards

The Employment Insurance card component is used to capture information that will impact EI calculations for employees.

Employment Insurance Card Component Hierarchy

The Employment Insurance card component has this shape:



The Employment Insurance card component uses the Employment Insurance flexfield context and data for this is loaded using the Component Detail record type. This card component has one override, which is supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to the *Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards* topic for the attributes to supply for the card component and component detail.

Card Component Name

When defining the Employment Insurance card component, specify a value of 'Employment Insurance' for the DirCardCompDefName attribute on the Card Component, Component Detail and Calculation Value Definition records.

Component Detail Flexfield Context

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context:

- Employment Insurance (HRX_CA_ORG_FEDERAL_EMPLOYMENT_INSURANCE)

Note: You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Calculation Value Definition Attributes

This is the only card component of the organization card that contains an override value.

The Calculation Value Definition record type specifies the value definition name for the override value.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName And, PayrollStatutoryUnitName Or, TaxRepUnitName	A unique identifier for the calculation value definition record. For new records, supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, LegislativeDataGroupName, DirCardDefinitionName, DirCardCompDefName And, PayrollStatutoryUnitName Or,	The parent EmploymentInsurance card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	TaxRepUnitName	
EffectiveStartDate	N/A	The start date of the parent card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The name of the card component this value definition is for. This is used to identify the value definition and should be supplied even when source keys are used to identify the record. Supply the same value as supplied to this attribute on the parent card component.
ValueDefinitionName	N/A	Specify 'Employer EI Rate'

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent Employment Insurance card component and a CalculationCard record for the owning Calculation Rules for Tax Reporting and Payroll Statutory Unit card.

Enterable Calculation Value Attributes

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record, which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName And, PayrollStatutoryUnitName Or, TaxRepUnitName	A unique identifier for the enterable calculation value record. For new records, supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName And, PayrollStatutoryUnitName Or, TaxRepUnitName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes.
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Related Topics

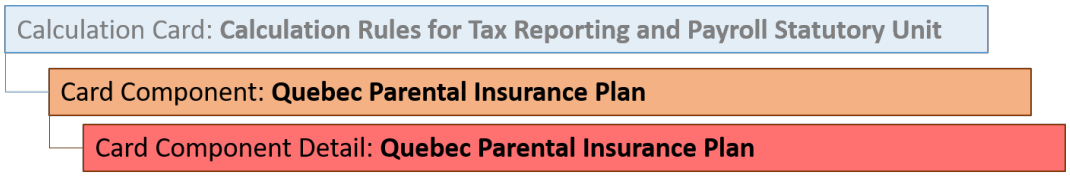
- Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards

Guidelines for Loading Quebec Parental Insurance Plan Card Components for Canadian Organization Cards

The Quebec Parental Insurance Plan card component is used to capture information that will impact QPIP calculations for employees. This component is only applicable to Quebec.

Quebec Parental Insurance Plan Card Component Hierarchy

The Quebec Parental Insurance Plan card component has this shape:



The Quebec Parental Insurance Plan card component uses the Quebec Parental Insurance Plan flexfield context and data for this is loaded using the Component Detail record type.

Refer to the *Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards* topic for the attributes to supply for each of these record types.

Card Component Name

When defining the Quebec Parental Insurance Plan card component, specify a value of 'Quebec Parental Insurance Plan' for the DirCardCompDefName attribute on the Card Component and Component Detail records.

In addition to the common card component attributes, you are required to supply the province information:

HCM Data Loader Attribute	Functional Description
Context1	The code of the province. Specify '19' for Quebec.

Component Detail Flexfield Context

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context:

- Quebec Parental Insurance Plan (HRX_CA_ORG_PROVINCE_PARENTAL_INSURANCE_PLAN)

Note: You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Related Topics

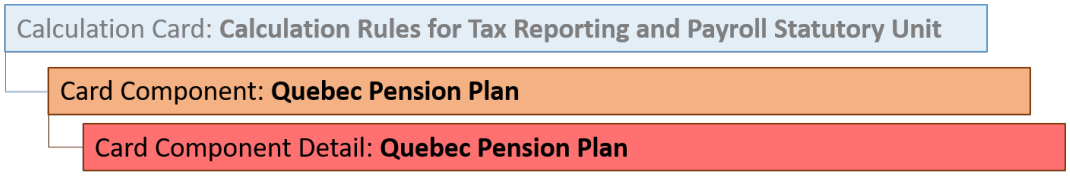
- Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards

Guidelines for Loading Quebec Pension Plan Card Components for Canadian Organization Cards

The Quebec Pension Plan card component is used to capture information that will impact QPP calculations for employees. This component is only applicable to Quebec.

Quebec Pension Plan Card Component Hierarchy

The Quebec Pension Plan card component has this shape:



The Quebec Pension Plan card component uses the Quebec Pension Plan flexfield context and data for this is loaded using the Component Detail record type.

Refer to the *Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards* topic for the attributes to supply for each of these record types.

Card Component Name

When defining the Quebec Pension Plan card component, specify a value of 'Quebec Pension Plan' for the DirCardCompDefName attribute on the Card Component and Component Detail records.

In addition to the common card component attributes, you are required to supply the province information:

HCM Data Loader Attribute	Functional Description
Context1	The code of the province. Specify '19' for Quebec.

Component Detail Flexfield Context

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context:

- Quebec Pension Plan (HRX_CA_ORG_PROVINCE_PENSION_PLAN)

Note: You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Related Topics

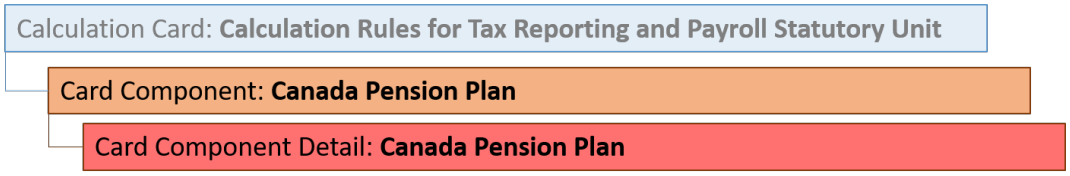
- Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards

Guidelines for Loading Canada Pension Plan Card Components for Canadian Organization Cards

The Canada Pension Plan card component is used to capture information that will impact CPP calculations for employees.

Canada Pension Plan Card Component Hierarchy

The Canada Pension Plan card component has this shape:



The **Canada Pension Plan** card component uses the **Canada Pension Plan** flexfield context and data for this is loaded using the Component Detail record type.

Refer to *Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards* topic for the attributes to supply for each of these record types.

Card Component Name

When defining the Canada Pension Plan card component, specify a value of 'Canada Pension Plan' for the DirCardCompDefName attribute on the Card Component and Component Detail records.

Component Detail Flexfield Context

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context:

- Canada Pension Plan (HRX_CA_ORG_FEDERAL_PENSION_PLAN)

Note: You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Related Topics

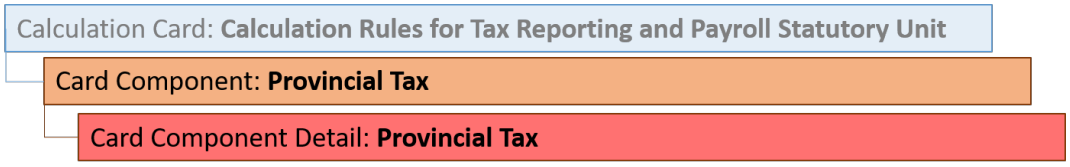
- Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards

Guidelines for Loading Provincial Tax Card Components for Canadian Organization Cards

The Provincial Tax card component is used to capture information that will impact Provincial Tax calculations for employees. This component is applicable to all provinces, except Quebec.

Provincial Tax Card Component Hierarchy

The Provincial Tax card component has this shape:



The Provincial Tax card component uses the Provincial Tax flexfield context and data for this is loaded using the Component Detail record type.

Refer to the *Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards* topic for the attributes to supply for each of these record types.

Card Component Name

When defining the Provincial Tax card component, specify a value of 'Provincial Tax' for the DirCardCompDefName attribute on the Card Component and Component Detail records.

In addition to the common card component attributes, you are required to supply the province information:

HCM Data Loader Attribute	Functional Description
Context1	The code of the province. Refer to the *Geography Codes for Canadian Provinces* topic.

Component Detail Flexfield Context

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context:

- Provincial Tax (HRX_CA_ORG_PROVINCE_INCOME_TAX)

Note: You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Related Topics

- Guidelines for Loading Canadian Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Geography Codes for Canadian Provinces

Provisional Medical and Workers Compensation

Overview of Provincial Medical Account Information for Canada

Provincial Medical is applicable only to the provinces that have provincial medical coverage funded by the employer.

Each province has different rules for processing the provincial medical liability. Those provinces are:

- British Columbia
- Ontario
- Manitoba
- Newfoundland and Labrador
- Quebec

The Provincial Medical Account is configured at the Payroll Statutory Unit (PSU) level. The details you define at the PSU level are valid for all the tax reporting units associated with the PSU.

The province of employment determines which Provincial Medical Account is associated to the employee's assignment. The default account is used to calculate the provincial medical liability, unless you override the Provincial Medical Account at the employee's assignment, department or location.

The Provincial Medical Carriers for each province are predefined in the Oracle Fusion Human Capital Management for Canada application.

You can configure overrides at the Department and Location level, as well as the employee assignment level.

For additional information on Provincial Medical Compensation, refer to the **Implementing Payroll for Canada** guide located at the Canada Information Center on My Oracle Support.

Canada Information Center:

https://support.oracle.com/rs?type=doc&id=2102586.2

- Canada tab > Product Documentation > Payroll Guides > Implementing Payroll for Canada

Related Topics

- Overview of Workers Compensation Account Information for Canada
- Guidelines for Loading Canadian Provincial Medical and Workers Compensation Account Information
- Guidelines for Loading Canadian Provincial Medical Account Overrides at the Location Level
- Guidelines for Loading Canadian Provincial Medical Account Overrides at the Department Level

Overview of Workers Compensation Account Information for Canada

Each province has different rules for Workers Compensation liability calculations. An employer can have multiple accounts in a province.

The Workers Compensation Account, and the Workers Compensation Classification Unit information is configured at the PSU level. The details you define at the PSU level are valid for all the tax reporting units associated with the PSU.

Since each province or board can have multiple accounts, you must designate one account as the default for the province or board.

Since multiple classification units can be defined for each account and each classification unit has an associated rate, you must designate one classification unit as the default for an account. There is a 1:1 relationship between the classification units and its associated rate.

The Workers Compensation Boards for each province are predefined in the Oracle Fusion Human Capital Management for Canada application.

You can configure overrides at the Job, Department and Location levels, as well as the employee assignment level.

For additional information on Workers Compensation, refer to the **Implementing Payroll for Canada** guide located at the Canada Information Center on My Oracle Support.

Canada Information Center:

https://support.oracle.com/rs?type=doc&id=2102586.2

- Canada tab > Product Documentation > Payroll Guides > Implementing Payroll for Canada

Related Topics

- Overview of Provincial Medical Account Information for Canada
- Guidelines for Loading Canadian Provincial Medical and Workers Compensation Account Information
- Guidelines for Loading Canadian Workers Compensation Overrides at the Job Level
- Guidelines for Loading Canadian Workers Compensation Overrides at the Location Level
- Guidelines for Loading Canadian Workers Compensation Overrides at the Department Level

Guidelines for Loading Canadian Provincial Medical and Workers Compensation Account Information

Provincial Medical and Workers Compensation account information is held against the Payroll Statutory Unit, using the Payroll Statutory Unit For Canada flexfield category.

Use the HCM Data Loader Organization object to maintain this data.

Considerations and Prerequisites

You must have previously defined the organizational structures for your business as required for HR and Payroll country-specific processes. For further information, please refer to the **Implementing Payroll for Canada** guide located at the Canada Information Center on My Oracle Support.

Canada Information Center:

https://support.oracle.com/rs?type=doc&id=2102586.2

- Canada tab > Product Documentation > Payroll Guides > Implementing Payroll for Canada

Create one Organization.dat file for each province. There are example files provided for each province that include both Provincial Medical and Workers Compensation data in one file. If your organization is not subject to Provincial Medical or Workers Compensation, you don't need to load the related data. Similarly, if you don't have any employees in Quebec, you don't need to load the file for Quebec. You only need to load the data that is relevant to your organization.

If you are loading overrides for Provincial Medical or Workers Compensation, additional files are required for the level the overrides are created at. Refer to the job, location, department and assignment topics.

Organization Record Types

The Organization object hierarchy has three components that must be supplied when loading Provincial Medical or Workers Compensation information:

Component	Functional Description	File Discriminator
Organization	The organization to update with Provincial Medical and Workers Compensation account details.	Organization
Organization Classification	The PSU details of the organization. This is required even if the organization exists and is already defined as a payroll statutory unit.	OrgUnitClassification
Organization Extra Information	Provincial Medical and Worker Compensation account details are held within flexfield segments and are loaded by the OrgInformation record.	OrgInformation

Organization Attributes

Supply one Organization record for each PSU you are providing Provincial Medical or Workers Compensation account details for.

The Organization record type requires these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the organization.
Name	The name of the payroll statutory unit.
ClassificationName	The name of the organization classification. Specify 'Payroll Statutory Unit'.

Organization Classification Attributes

Supply one OrgUnitClassification record for each PSU you have defined. This is used to capture the PSU details for which to capture information held at the PSU level.

The OrgUnitClassification record type requires these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the payroll statutory unit classification.
OrganizationName	The name of the payroll statutory unit. This must match the value supplied to the Name attribute on the parent Organization record.
ClassificationName	The name of the organization classification. Specify 'Payroll Statutory Unit'.
LegislationCode	The code of the legislation. Specify 'CA'.
CategoryCode	The category code of the classification. Specify 'HCM_PSU_CA'.
Status	The status of the classification. Specify 'A' for Active or 'I' for Inactive.

Organization Extra Information Attributes

Supply OrgInformation records for each combination of province/Workers Compensation and province/Provincial Medical combination, where applicable. This is used to capture the account and rate details for Provincial Medical and Workers Compensation, at the PSU level.

Supply details in this record for the PER_ORGANIZATION_INFORMATION_EFF flexfield context.

The OrgInformation component record uses these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the payroll statutory unit information.
OrganizationName	The name of the payroll statutory unit. This must match the value supplied to the Name attribute on the parent Organization record.
ClassificationName	The name of the organization classification. Specify 'Payroll Statutory Unit'
LegislationCode	The code of the legislation. Specify 'CA'.
SequenceNumber	The sequence number for the classification component record.

HCM Data Loader Attribute	Functional Description
	Note: This is sequential according to the component context. For example, if loading 2 provincial medical records and 1 for workers compensation, populate 1 and 2 for the provincial medical records and 1 for the workers compensation record.
EFF_CATEGORY_CODE	The Payroll Statutory Unit For Canada category code for the extensible flexfield for the component record. Specify 'HCM_PSU_CA'.
FLEX:PER_ORGANIZATION_INFORMATION_EFF	The organization flexfield context for the province. Specify one of the flexfield contexts listed in the tables below.
OrgInformationContext	Supply the same value as the FLEX:PER_ORGANIZATION_INFORMATION_EFF attribute value.

Provincial Medical Account Flexfield Contexts

Provincial Medical is applicable only to the provinces that have provincial medical coverage funded by the employer. The province determines the flexfield code to use for provincial medical data.

The province determines the flexfield code to use:

Province	Flexfield Context Name	Flexfield Context Code
Flexfield Context Code	British Columbia Provincial Medical Account Details	ORA_HRX_CA_PSU_PM_BC_DETAIL
Manitoba	Manitoba Provincial Medical Account Details	ORA_HRX_CA_PSU_PM_MB_DETAIL
Newfoundland and Labrador	Newfoundland and Labrador Provincial Medical Account Details	ORA_HRX_CA_PSU_PM_NL_DETAIL
Ontario	Ontario Provincial Medical Account Details	ORA_HRX_CA_PSU_PM_ON_DETAIL
Quebec	Quebec Provincial Medical Account Details	ORA_HRX_CA_PSU_PM_QC_DETAIL

Workers Compensation Flexfield Contexts

The province determines the flexfield code to use for Workers Compensation data:

Province	Flexfield Context Name	Flexfield Context Code
Alberta	Alberta Account Details	ORA_HRX_CA_PSU_WC_AB_DETAIL
British Columbia	British Columbia Account Details	ORA_HRX_CA_PSU_WC_BC_DETAIL
Manitoba	Manitoba Account Details	ORA_HRX_CA_PSU_WC_MB_DETAIL
New Brunswick	New Brunswick Account Details	ORA_HRX_CA_PSU_WC_NB_DETAIL
Newfoundland and Labrador	Newfoundland and Labrador Account Details	ORA_HRX_CA_PSU_WC_NL_DETAIL
Nova Scotia	Nova Scotia Account Details	ORA_HRX_CA_PSU_WC_NS_DETAIL
Northwest Territories	Northwest Territories Account Details	ORA_HRX_CA_PSU_WC_NT_DETAIL
Nunavut	Nunavut Account Details	ORA_HRX_CA_PSU_WC_NU_DETAIL

Province	Flexfield Context Name	Flexfield Context Code
Ontario	Ontario Account Details	ORA_HRX_CA_PSU_WC_ON_DETAIL
	Ontario Workers Compensation	ORA_HRX_CA_PSU_WC_ON_HEADER
Prince Edward Island	Prince Edward Island Account Details	ORA_HRX_CA_PSU_WC_PE_DETAIL
Quebec	Quebec Account Details	ORA_HRX_CA_PSU_WC_QC_DETAIL
Saskatchewan	Saskatchewan Account Details	ORA_HRX_CA_PSU_WC_SK_DETAIL
Yukon	Yukon Account Details	ORA_HRX_CA_PSU_WC_YT_DETAIL

Note: There are two flexfield contexts available for Ontario.

Flexfield Segment Attributes

Use the View Business Objects task, navigating to the Flexfield Attributes tab to review the list of attributes supported for each flexfield context. These can be summarised as follows:

Provincial Medical Account Flexfield Segment Attributes

Change the Xx code within the attribute name and flexfield context to the province code. For example, oraHrxCaBcPmUnique(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_BC_DETAIL) for British Columbia.

HCM Data Loader Attribute	Functional Description
oraHrxCaXxPmUnique(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_PM_XX_DETAIL)	The unique number identifier of the Provincial Medical flexfield data. Enter a unique number for each record of this type.
oraHrxCaXxPmAcctNumber(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_PM_XX_DETAIL)	The Provincial Medical Account Number associated with the Provincial Medical Carrier.
oraHrxCaXxPmAcctName(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_PM_XX_DETAIL)	The Provincial Medical Account Name associated with the Provincial Medical Carrier.
oraHrxCaXxPmDefaultAcct(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_PM_XX_DETAIL)	The default account for the province or the Provincial Medical Carrier. Specify 'Y' or 'N'. **Note:** Only one account per province or carrier can be set as the default.
oraHrxCaXxPmStartAnnPay(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_PM_XX_DETAIL)	The Starting Annual Payroll threshold.
oraHrxCaXxPmEndAnnPay(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_PM_XX_DETAIL)	The Ending Annual Payroll threshold.
oraHrxCaXxPmRate(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_PM_XX_DETAIL)	The rate used to calculate the Provincial Medical liability for the account.

HCM Data Loader Attribute	Functional Description
	Note: Certain employees may be exempt from Provincial Medical premiums. The exempt status is set at the assignment level to ensure that provincial medical calculations are not processed for the exempted employees.
oraCalcLaibPayroll(ORA_HRX_CA_PSU_PM_ON_HEADER)	The option to calculate the liability in the payroll run for Ontario. Specify 'Y' or 'N'. Leave this attribute out if you are not loading data for Ontario. **Note:** This attribute is only applicable to Ontario.

Note: For the province of Quebec we capture only the rate. The account number and name are not required for Quebec as the provincial medical account number is the Quebec Identification Number (QIN). The payment range details are read-only, with the Starting Annual Payroll set to 0 and the Ending Annual Payroll set to 999,999,999,999.99. The value you enter in the Rate field is used to calculate the employer liability.

Workers Compensation

Change the Xx code within the attribute name and flexfield context to the province code. For example, oraHrxCaBcUnique(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_BC_DETAIL) for British Columbia.

HCM Data Loader Attribute	Functional Description
oraHrxCaXxUnique(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_WC_XX_DETAIL)	The unique number identifier of the Workers Compensation flexfield data. Enter a unique number for each record of this type.
oraHrxCaXxAcctNumber(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_WC_XX_DETAIL)	The Workers Compensation Account Number associated with the Workers Compensation Board.
OraHrxCaXxAcctName(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_WC_XX_DETAIL)	The Workers Compensation Account Name associated with the Workers Compensation Board.
OraHrxCaXxDefault(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_WC_XX_DETAIL)	The default account for the province or the Workers Compensation Board. Specify 'Y' or 'N'. **Note:** Only one account per province or board can be set as the default.
oraHrxCaXxClUnit(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_WC_XX_DETAIL)	The Classification Unit for the Workers Compensation Account.
oraHrxCaXxClUnitDesc(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_WC_XX_DETAIL)	The Classification Unit Description for the Workers Compensation Account.
oraHrxCaXxRate(PER_ORGANIZATION_INFORMATION_EFF= ORA_HRX_CA_PSU_WC_XX_DETAIL)	The rate used to calculate the Workers Compensation liability for the Classification Unit. **Note:** A Classification Unit cannot have two different rates for the same province. If a province has two different rates, you must define two different Classification Units.

HCM Data Loader Attribute	Functional Description
oraHrxCaXxDefaultRate(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_XX_DETAIL)	The default rate for the Workers Compensation Account. Specify 'Y' or 'N'. **Note:** Only one rate can be set as the default per account.

Related Topics

- Overview of Provincial Medical Account Information for Canada
- Overview of Workers Compensation Account Information for Canada
- Example of Loading Provincial Medical and Worker Compensation for British Columbia
- Example of Loading Provincial Medical and Worker Compensation for Ontario
- Example of Loading Provincial Medical and Worker Compensation for Quebec

Example of Loading Provincial Medical and Worker Compensation for British Columbia

The following Organization.dat file uploads both Provincial Medical and Worker Compensation information against an existing PSU for British Columbia.

A separate OrgInformation record is supplied for each flexfield context:

Provincial Medical:

- British Columbia Provincial Medical Account Details (ORA_HRX_CA_PSU_PM_BC_DETAIL)

Workers Compensation:

- British Columbia Account Details (ORA_HRX_CA_PSU_WC_BC_DETAIL)

```
METADATA|Organization|EffectiveStartDate|Name|ClassificationName
MERGE|Organization|2001/01/01|CA PSU|Payroll Statutory Unit

METADATA|OrgUnitClassification|EffectiveStartDate|OrganizationName|ClassificationName|LegislationCode|
CategoryCode|Status
MERGE|OrgUnitClassification|2001/01/01|CA PSU|Payroll Statutory Unit|CA|HCM_PSU_CA|A

METADATA|OrgInformation|EffectiveStartDate|OrganizationName|ClassificationName|LegislationCode|
SequenceNumber|EFF_CATEGORY_CODE|FLEX:PER_ORGANIZATION_INFORMATION_EFF|OrgInformationContext|
oraHrxCaBcUnique(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_BC_DETAIL)|
oraHrxCaBcAcctNumber(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_BC_DETAIL)|
OraHrxCaBcAcctName(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_BC_DETAIL)|
OraHrxCaMbDefault(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_BC_DETAIL)|
oraHrxCaBcClUnit(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_BC_DETAIL)|
oraHrxCaBcClUnitDesc(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_BC_DETAIL)|
oraHrxCaBcRate(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_BC_DETAIL)|
oraHrxCaBcDefaultRate(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_BC_DETAIL)|
oraHrxCaBcPmUnique(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_BC_DETAIL)|
oraHrxCaBcPmAcctNumber(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_BC_DETAIL)|
oraHrxCaBcPmAcctName(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_BC_DETAIL)|
oraHrxCaBcPmDefaultAcct(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_BC_DETAIL)|
oraHrxCaBcPmStartAnnPay(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_BC_DETAIL)|
oraHrxCaBcPmEndAnnPay(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_BC_DETAIL)|
oraHrxCaBcPmRate(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_BC_DETAIL)
```

```
MERGE|OrgInformation|2001/01/01|CA PSU|Payroll Statutory Unit|CA|1|HCM_PSU_CA|ORA_HRX_CA_PSU_WC_BC_DETAIL|
ORA_HRX_CA_PSU_WC_BC_DETAIL|1|BCWC0001|BC WC Account1|Y|BCCU0001|Classification Unit1|1|Y|||||||
MERGE|OrgInformation|2001/01/01|CA PSU|Payroll Statutory Unit|CA|1|HCM_PSU_CA|ORA_HRX_CA_PSU_PM_BC_DETAIL|
ORA_HRX_CA_PSU_PM_BC_DETAIL|||||||||1|BCPM00001|BC PM Account 1|Y|0|999999999999.99|4
```

Example of Loading Provincial Medical and Worker Compensation for Ontario

The following Organization.dat file uploads both Provincial Medical and Worker Compensation information against an existing PSU for Ontario.

A separate OrgInformation record is supplied for each flexfield context:

Provincial Medical:

- Ontario Provincial Medical Account Details (ORA_HRX_CA_PSU_PM_ON_DETAIL)

Workers Compensation:

- Ontario Account Details (ORA_HRX_CA_PSU_WC_ON_DETAIL)
- Ontario Workers Compensation (ORA_HRX_CA_PSU_PM_ON_HEADER)

The flexfield attribute oraCalcLaibPayroll, available on the Ontario Workers Compensation flexfield context is only required for Ontario. It's an option to calculate the liability in the payroll run for Ontario, with valid values of Y or N.

```
METADATA|Organization|EffectiveStartDate|Name|ClassificationName
MERGE|Organization|2001/01/01|CA PSU|Payroll Statutory Unit

METADATA|OrgUnitClassification|EffectiveStartDate|OrganizationName|ClassificationName|LegislationCode|
CategoryCode|Status
MERGE|OrgUnitClassification|2001/01/01|CA PSU|Payroll Statutory Unit|CA|HCM_PSU_CA|A

METADATA|OrgInformation|EffectiveStartDate|OrganizationName|ClassificationName|LegislationCode|
EFF_CATEGORY_CODE|FLEX:PER_ORGANIZATION_INFORMATION_EFF|OrgInformationContext|SequenceNumber|
OrgUnitClassificationId|oraHrxCaOnUnique(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_ON_DETAIL)|
oraHrxCaOnAcctNumber(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_ON_DETAIL)|
oraHrxCaOnAcctName(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_ON_DETAIL)|
oraHrxCaOnDefault(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_ON_DETAIL)|
oraHrxCaOnClUnit(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_ON_DETAIL)|
oraHrxCaOnClUnitDesc(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_ON_DETAIL)|
oraHrxCaOnRate(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_ON_DETAIL)|
oraHrxCaOnDefaultRate(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_ON_DETAIL)|
oraHrxCaOnPmUnique(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_ON_DETAIL)|
oraHrxCaOnPmAcctNumber(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_ON_DETAIL)|
oraHrxCaOnPmAcctName(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_ON_DETAIL)|
oraHrxCaOnPmDefaultAcct(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_ON_DETAIL)|
oraHrxCaOnPmStartAnnPay(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_ON_DETAIL)|
oraHrxCaOnPmEndAnnPay(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_ON_DETAIL)|
oraHrxCaOnPmRate(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_ON_DETAIL)|
oraCalcLaibPayroll(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_ON_HEADER)
MERGE|OrgInformation|2001/01/01|CA PSU|Payroll Statutory Unit|CA|HCM_PSU_CA|ORA_HRX_CA_PSU_WC_ON_DETAIL|
ORA_HRX_CA_PSU_WC_ON_DETAIL|1||1|ONWC0001|ON WC Account1|Y|ONCU0001|Classification Unit1|1|Y|||||||
MERGE|OrgInformation|2001/01/01|CA PSU|Payroll Statutory Unit|CA|HCM_PSU_CA|ORA_HRX_CA_PSU_PM_ON_DETAIL|
ORA_HRX_CA_PSU_PM_ON_DETAIL|1|||||||||||1|ONPM0001|ON PM Account1|Y|0|999999999999.99|4|
MERGE|OrgInformation|2001/01/01|CA PSU|Payroll Statutory Unit|CA|HCM_PSU_CA|ORA_HRX_CA_PSU_PM_ON_HEADER|
ORA_HRX_CA_PSU_PM_ON_HEADER|1|||||||||||||||||Y
```

Example of Loading Provincial Medical and Worker Compensation for Quebec

The following Organization.dat file uploads both Provincial Medical and Worker Compensation information against an existing PSU for Quebec.

A separate OrgInformation record is supplied for each flexfield context:

Provincial Medical:

- Quebec Provincial Medical Account Details (ORA_HRX_CA_PSU_PM_QC_DETAIL)

Workers Compensation:

- Quebec Account Details (ORA_HRX_CA_PSU_WC_QC_DETAIL)

Note: The Provincial Medical Account Number and Provincial Medical Account Name are not captured for the Quebec Provincial Medical Account Details flexfield context.

```
METADATA|Organization|EffectiveStartDate|Name|ClassificationName
MERGE|Organization|2001/01/01|CA PSU|Payroll Statutory Unit

METADATA|OrgUnitClassification|EffectiveStartDate|OrganizationName|ClassificationName|LegislationCode|
CategoryCode|Status
MERGE|OrgUnitClassification|2001/01/01|CA PSU|Payroll Statutory Unit|CA|HCM_PSU_CA|A

METADATA|OrgInformation|EffectiveStartDate|OrganizationName|ClassificationName|LegislationCode|
SequenceNumber|EFF_CATEGORY_CODE|FLEX:PER_ORGANIZATION_INFORMATION_EFF|OrgInformationContext|
oraHrxCaQcUnique(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_QC_DETAIL)|
oraHrxCaQcAcctNumber(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_QC_DETAIL)|
oraHrxCaQcAcctName(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_QC_DETAIL)|
oraHrxCaQcDefault(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_QC_DETAIL)|
oraHrxCaQcClUnit(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_QC_DETAIL)|
oraHrxCaQcClUnitDesc(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_QC_DETAIL)|
oraHrxCaQcRate(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_QC_DETAIL)|
oraHrxCaQcDefaultRate(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_WC_QC_DETAIL)|
oraHrxCaQcPmStartAnnPay(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_QC_DETAIL)|
oraHrxCaQcPmEndAnnPay(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_QC_DETAIL)|
oraHrxCaQcPmRate(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PSU_PM_QC_DETAIL)
MERGE|OrgInformation|2001/01/01|CA PSU|Payroll Statutory Unit|CA|1|HCM_PSU_CA|ORA_HRX_CA_PSU_WC_QC_DETAIL|
ORA_HRX_CA_PSU_WC_QC_DETAIL|1|QCWC0001|QC WC Account1|Y|QCCU0001|Classification Unit1|1|Y|||
MERGE|OrgInformation|2001/01/01|CA PSU|Payroll Statutory Unit|CA|1|HCM_PSU_CA|ORA_HRX_CA_PSU_PM_QC_DETAIL|
ORA_HRX_CA_PSU_PM_QC_DETAIL|||||||||0|999999999999.99|4
```

Guidelines for Loading Canadian Workers Compensation Overrides at the Job Level

Workers Compensation overrides at the job level are created using the Job HCM Data Loader business object.

To upload Workers Compensation overrides these components of the Job object hierarchy are used:

Component	Functional Description	File Discriminator
Job	The job to override Workers Compensation data for.	Job
Job Legislative Extra Information	Worker Compensation details are held within flexfield segments and are loaded by the JobLegislative record.	JobLegislative

Job Attributes

Supply one Job record for each job to provide overrides for.

The Job record uses these attributes to identify the existing job you are updating:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the job.
SetCode	The set code of the job.
JobCode	The code that identifies the job.

Job Legislative Extra Information Attributes

Supply one JobLegislative component record for each job to provide overrides for. These are held on the PER_JOBS_LEG_EFF flexfield.

Workers Compensation overrides at the job level are held on the on JOB_LEG flexfield category with this flexfield context:

- Canada Workers Compensation Job Information (ORA_HRX_CA_WC_JOBS)

Supply these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the Workers Compensation override information.
SetCode	The set code of the job. Specify the same value as in the SetCode of the Job component record.
JobCode	The code that identifies the job. Specify the same value as in the JobCode of the Job component record.
LegislationCode	The code of the legislation. Specify ‘CA’.
SequenceNumber	The sequence number for the component record. Enter a unique number for each record of this type. If multiple rows for the same type, increase each row number by 1.

HCM Data Loader Attribute	Functional Description
EFF_CATEGORY_CODE	The flexfield context category code that is used to supply the Workers Compensation overrides. Specify 'JOB_LEG'.
FLEX:PER_JOBS_LEG_EFF	The flexfield context code, specify 'ORA_HRX_CA_WC_JOBS'
InformationCategory	The flexfield context code, specify 'ORA_HRX_CA_WC_JOBS'
oraHrxCaWcJobUniqueSeg(PER_JOBS_LEG_EFF= ORA_HRX_CA_WC_JOBS)	The sequence number for the component record. Enter a unique number for each record of this type. If multiple rows for the same type, increase each row number by 1.
oraHrxCaWcJobPsu(PER_JOBS_LEG_EFF= ORA_HRX_CA_WC_JOBS)	The ID used to identify the Workers Compensation Payroll Statutory Unit override value of the component record.
oraHrxCaWcJobProv(PER_JOBS_LEG_EFF= ORA_HRX_CA_WC_JOBS)	The geocode that identifies the Workers Compensation Province override value.
oraHrxCaWcJobClUnit(PER_JOBS_LEG_EFF= ORA_HRX_CA_WC_JOBS)	The Workers Compensation Classification Unit override value of the component record. **Note:** This is a combination of the Workers Compensation Account Number and Classification Unit value, concatenated with an underscore value (for example, <Account Number>_<Classification Unit>).

Note: Refer to the Cloud Customer Connect topic: *BI Publisher Report: Canada Workers Compensation Classification Units for Overrides* for a report that extracts the values required for the PSU, Province and Classification Unit flexfield segment attributes.

Example of Loading Workers Compensation Overrides against a Job for Alberta

This Job.dat file will update the existing SALES_CONS job to define an override for Alberta.

```
METADATA|Job|EffectiveStartDate|SetCode|JobCode
MERGE|Job|2000/01/01|COMMON|SALES_CONS

METADATA|JobLegislative|EffectiveStartDate|SetCode|JobCode|LegislationCode|SequenceNumber|EFF_CATEGORY_CODE|
FLEX:PER_JOBS_LEG_EFF|InformationCategory|oraHrxCaWcJobUniqueSeg(PER_JOBS_LEG_EFF=ORA_HRX_CA_WC_JOBS)|
oraHrxCaWcJobPsu(PER_JOBS_LEG_EFF=ORA_HRX_CA_WC_JOBS)|
oraHrxCaWcJobProv(PER_JOBS_LEG_EFF=ORA_HRX_CA_WC_JOBS)|
oraHrxCaWcJobClUnit(PER_JOBS_LEG_EFF=ORA_HRX_CA_WC_JOBS)
MERGE|JobLegislative|2000/01/01|COMMON|SALES_CONS|CA|1|JOB_LEG|ORA_HRX_CA_WC_JOBS|ORA_HRX_CA_WC_JOBS|1|
300100083566799|1|AB1234_ABCU1234
```

Guidelines for Loading Canadian Workers Compensation Overrides at the Location Level

Workers Compensation overrides at the location level are created using the Location HCM Data Loader business object.

Component	Functional Description	File Discriminator
Location	The location to override Workers Compensation data for.	Location
Location Legislative Extra Information	Worker Compensation details are held within flexfield segments and are loaded by the LocationLegislative record.	LocationLegislative

Location Attributes

Supply one Location record for each location to provide overrides for.

The Location record uses these attributes to identify the existing location you are updating:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the location override.
SetCode	The set code of the location override. The default is 'COMMON'.
LocationCode	The code that identifies the location override.

Location Legislative Extra Information Attributes

Supply one LocationLegislative component record for each location to provide overrides for. These are held on the PER_LOCATION_LEG_EFF flexfield.

The attributes to supply are different for Provincial Medical and Workers Compensation overrides.

Workers Compensation

Workers Compensation overrides at the location level are held on the on HCM_LOC_LEG flexfield category with this flexfield context:

- Canada Workers Compensation Location Information (ORA_HRX_CA_WC_LOC)

Supply these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the Workers Compensation override information.
SetCode	The set code of the location override. Specify the same value as in the SetCode of the Location component record.
LocationCode	The code that identifies the location. Specify the same value as in the LocationCode of the Location component record.
LegislationCode	The code of the legislation. Specify ‘CA’.

HCM Data Loader Attribute	Functional Description
EFF_CATEGORY_CODE	The flexfield context category code that is used to supply the Workers Compensation overrides. Specify 'HCM_LOC_LEG'.
FLEX:PER_JOBS_LEG_EFF	The flexfield context code, specify 'ORA_HRX_CA_WC_LOC'
InformationCategory	The flexfield context code, specify 'ORA_HRX_CA_WC_LOC'
SequenceNumber	The sequence number for the component record. Enter a unique number for each record of this type. If multiple rows for the same type, increase each row number by 1. **Note:** This is sequential according to the component context. For example, if loading 2 provincial medical records and 1 for workers compensation, populate 1 and 2 for the provincial medical records and 1 for the workers compensation record.
oraHrxCaWcLocUniqueSeg(PER_LOCATION_LEG_EFF= ORA_HRX_CA_WC_LOC)	The sequence number for the component record. Enter a unique number for each record of this type. If multiple rows for the same type, increase each row number by 1.
oraHrxCaWcLocPsu(PER_LOCATION_LEG_EFF= ORA_HRX_CA_WC_LOC)	The ID used to identify the Payroll Statutory Unit override value of the component record.
oraHrxCaWcLocClUnit(PER_LOCATION_LEG_EFF= ORA_HRX_CA_WC_LOC)	The Workers Compensation Classification Unit override value of the component record. **Note:** This is a combination of the Workers Compensation Account Number and Classification Unit value, concatenated with an underscore value (for example, <Account Number>_<Classification Unit>).

Note: Refer to the Cloud Customer Connect topic: *BI Publisher Report: Canada Workers Compensation Classification Units for Overrides* for a report that extracts the values required for the PSU and Classification Unit flexfield segment attributes.

Example of Loading Workers Compensation Overrides against a Location for Canada

This Location.dat file will update the existing Alberta location to define the Workers Compensation Classification Unit override.

```
METADATA|Location|EffectiveStartDate|SetCode|LocationCode
MERGE|Location|2018/01/01|COMMON|Alberta, CA

METADATA|LocationLegislative|EffectiveStartDate|SetCode|LocationCode|
LegislationCode|EFF_CATEGORY_CODE|FLEX:PER_LOCATION_LEG_EFF|LleInformationCategory|
SequenceNumber|oraHrxCaWcLocUniqueSeg(PER_LOCATION_LEG_EFF=ORA_HRX_CA_WC_LOC)|
oraHrxCaWcLocPsu(PER_LOCATION_LEG_EFF=ORA_HRX_CA_WC_LOC)|
oraHrxCaWcLocClUnit(PER_LOCATION_LEG_EFF=ORA_HRX_CA_WC_LOC)
MERGE|LocationLegislative|2018/01/01|COMMON|Alberta, CA|CA|HCM_LOC_LEG|ORA_HRX_CA_WC_LOC|ORA_HRX_CA_WC_LOC|
1|1|300100083566799|AB1234_ABCU1234
```

Guidelines for Loading Canadian Provincial Medical Account Overrides at the Location Level

Provincial Medical account overrides at the location level are created using the Location HCM Data Loader business object.

To upload Provincial Medical account overrides these components of the Location object hierarchy are used:

Component	Functional Description	File Discriminator
Location	The location to override Provincial Medical data for.	Location
Location Legislative Extra Information	Provincial Medical Account details are held within flexfield segments and are loaded by the LocationLegislative record.	LocationLegislative

Location Attributes

Supply one Location record for each location to provide overrides for. The Location record uses these attributes to identify the existing location you are updating:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the location override.
SetCode	The set code of the location override. The default is 'COMMON'.
LocationCode	The code that identifies the location override.

Location Legislative Extra Information Attributes

Supply one LocationLegislative component record for each location to provide overrides for. These are held on the PER_LOCATION_LEG_EFF flexfield.

The attributes to supply are different for Provincial Medical and Workers Compensation overrides.

Provincial Medical:

Provincial Medical account overrides at the location level are held on the on HCM_LOC_LEG flexfield category with this flexfield context:

- Canada Provincial Medical Location Information (ORA_HRX_CA_PM_LOC)

Supply these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the Workers Compensation override information.

HCM Data Loader Attribute	Functional Description
SetCode	The set code of the location override. Specify the same value as in the SetCode of the Location component record.
JobCode	The code that identifies the overridden location. Specify the same value as in the LocationCode of the Location component record.
LegislationCode	The code of the legislation. Specify 'CA'.
EFF_CATEGORY_CODE	The flexfield context category code that is used to supply the Workers Compensation overrides. Specify 'HCM_LOC_LEG'
FLEX:PER_JOBS_LEG_EFF	The flexfield context code, specify 'ORA_HRX_CA_PM_LOC'
InformationCategory	The flexfield context code, specify 'ORA_HRX_CA_PM_LOC'
SequenceNumber	The sequence number for the component record. Enter a unique number for each record of this type. If multiple rows for the same type, increase each row number by 1. **Note:** This is sequential according to the component context. For example, if loading 2 provincial medical records and 1 for workers compensation, populate 1 and 2 for the provincial medical records and 1 for the workers compensation record.
oraHrxCaWcLocUniqueSeg(PER_LOCATION_LEG_EFF= ORA_HRX_CA_WC_LOC)	The sequence number for the component record. Enter a unique number for each record of this type. If multiple rows for the same type, increase each row number by 1.
oraHrxCaWcLocPsu(PER_LOCATION_LEG_EFF= ORA_HRX_CA_WC_LOC)	The ID used to identify the Payroll Statutory Unit override value of the component record.
oraHrxCaWcLocClUnit(PER_LOCATION_LEG_EFF= ORA_HRX_CA_WC_LOC)	The Workers Compensation Classification Unit override value of the component record. **Note:** This is a combination of the Workers Compensation Account Number and Classification Unit value, concatenated with an underscore value (for example, <Account Number>_<Classification Unit>).

Note: Refer to the Cloud Customer Connect topic: *BI Publisher Report: Canada Provincial Medical Accounts for Overrides* for a report that extracts the values required for the PSU and account flexfield segment attributes..

Example of Loading Provincial Medical Overrides against a Location

This Location.dat file will update the existing Manitoba location to define the Provincial Medical Account override.

```
METADATA|Location|EffectiveStartDate|SetCode|LocationCode
MERGE|Location|2018/01/01|COMMON|Manitoba, CA

METADATA|LocationLegislative|EffectiveStartDate|SetCode|LocationCode|
LegislationCode|EFF_CATEGORY_CODE|FLEX:PER_LOCATION_LEG_EFF|LleInformationCategory|
SequenceNumber|oraHrxCaPmLocUniqueSeg(PER_LOCATION_LEG_EFF=ORA_HRX_CA_PM_LOC)|
oraHrxCaPmLocPsu(PER_LOCATION_LEG_EFF=ORA_HRX_CA_PM_LOC)|
oraHrxCaPmLocAccount(PER_LOCATION_LEG_EFF=ORA_HRX_CA_PM_LOC)
MERGE|LocationLegislative|2018/01/01|COMMON|Manitoba, CA|CA|HCM_LOC_LEG|ORA_HRX_CA_PM_LOC|ORA_HRX_CA_PM_LOC|
1|1|300100023301253|MBP22222
```

Guidelines for Loading Canadian Workers Compensation Overrides at the Department Level

Workers Compensation overrides at the department level are created using the Organization HCM Data Loader business object.

To upload Workers Compensation overrides, these components of the Organization object hierarchy are used:

Component	Functional Description	File Discriminator
Organization	The department to override Workers Compensation data for.	Organization
Organization Classification	The organization classification. This is required even if the organization exists and is already defined as a department.	OrgUnitClassification
Organization Extra Information	Worker Compensation account details are held within flexfield segments and are loaded by the OrgInformation record.	OrgInformation

Organization Attributes

Supply one Organization record for each department you are providing Provincial Medical account override details for.

The Organization record type requires these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the department override.
Name	The name of the department override.
ClassificationName	The name of the organization classification. Specify 'DEPARTMENT'

Organization Classification Attributes

Supply one OrgUnitClassification record for each department you are providing Workers Compensation override details for. The OrgUnitClassification record type requires these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the department classification override.
OrganizationName	The name of the department to override. This must match the value supplied to the Name attribute on the parent Organization record.
ClassificationName	The name of the department classification. Specify 'DEPARTMENT'.

Organization Extra Information Attributes

Supply one OrgInformation record for each department to provide overrides for. These are held in the PER_ORGANIZATION_INFORMATION_EFF flexfield.

The attributes to supply are different for Provincial Medical and Workers Compensation overrides.

Workers Compensation:

Workers Compensation overrides at the department level are held on the on DEPARTMENT flexfield category with this flexfield context:

- Canada Workers Compensation Department Information (ORA_HRX_CA_WC_DEPT)

Supply these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the department override information.
OrganizationName	The name of the department. This must match the value supplied to the Name attribute on the parent Organization record.
ClassificationCode	The classification code of the component record for the override. Specify 'DEPARTMENT'.
ClassificationName	The name of the organization classification. Specify 'DEPARTMENT'.
SequenceNumber	The sequence number for the classification component record. Enter a unique number for each record of this type. If multiple rows for the same type, increase each row number by 1. **Note:** This is sequential according to the component context. For example, if loading 2 provincial medical records and 1 for workers compensation, populate 1 and 2 for the provincial medical records and 1 for the workers compensation record.
EFF_CATEGORY_CODE	The flexfield context category code that is used to supply the Workers Compensation overrides. Specify 'DEPARTMENT'.
FLEX:PER_ORGANIZATION_INFORMATION_EFF	The flexfield context code that is used to supply the Workers Compensation overrides. Specify 'ORA_HRX_CA_PM_DEPT'.
OrgInformationContext	Supply the same value as the FLEX:PER_ORGANIZATION_INFORMATION_EFF attribute value.
oraHrxCaWcDeptUniqueSeg(PER_ORGANIZATION_INFORMATION_EFF =ORA_HRX_CA_WC_DEPT)	The sequence number for the component record. Enter a unique number for each record of this type. If multiple rows for the same type, increase each row number by 1.
oraHrxCaWcDeptPsu(PER_ORGANIZATION_INFORMATION_EFF =ORA_HRX_CA_WC_DEPT)	The ID used to identify the Payroll Statutory Unit.
oraHrxCaWcDeptProv(PER_ORGANIZATION_INFORMATION_EFF =ORA_HRX_CA_WC_DEPT)	The geocode that identifies the Workers Compensation Province override value.
oraHrxCaWcDeptClUnit(PER_ORGANIZATION_INFORMATION_EFF =ORA_HRX_CA_WC_DEPT)	The Workers Compensation Classification Unit override value of the component record. **Note:** For the Workers Compensation override value, this is a combination of the Workers Compensation Account Number and Classification Unit value, concatenated with an underscore value (for example, <Account Number>_<Classification Unit>).

HCM Data Loader Attribute	Functional Description

Note: Refer to the Cloud Customer Connect topic: *BI Publisher Report: Canada Workers Compensation Classification Units for Overrides* for a report that extracts the values required for the PSU, Province and Classification Unit flexfield segment attributes.

Example of Loading Workers Compensation Overrides against a Department for Canada

This Organization.dat file will update the existing Administration department to define the Workers Compensation Classification Unit override.

```
METADATA|Organization|EffectiveStartDate|Name|ClassificationName
MERGE|Organization|2016/01/01|Administration|DEPARTMENT

METADATA|OrgUnitClassification|EffectiveStartDate|OrganizationName|ClassificationName
MERGE|OrgUnitClassification|2016/01/01|Administration|DEPARTMENT

METADATA|OrgInformation|EffectiveStartDate|OrganizationName|ClassificationName|
EFF_CATEGORY_CODE|FLEX:PER_ORGANIZATION_INFORMATION_EFF|OrgInformationContext|
SequenceNumber|oraHrxCaWcDeptUniqueSeg(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_WC_DEPT)|
oraHrxCaWcDeptPsu(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_WC_DEPT)|
oraHrxCaWcDeptProv(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_WC_DEPT)|
oraHrxCaWcDeptClUnit(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_WC_DEPT)
MERGE|OrgInformation|2018/01/01|Administration|DEPARTMENT|DEPARTMENT|ORA_HRX_CA_WC_DEPT|ORA_HRX_CA_WC_DEPT|
1|1|300100083566799|1|AB1234_ABCU1234
```

Guidelines for Loading Canadian Provincial Medical Account Overrides at the Department Level

Provincial Medical account overrides at the department level are created using the Organization HCM Data Loader business object.

To upload Provincial Medical account overrides, these components of the Organization object hierarchy are used:

Component	Functional Description	File Discriminator
Organization	The department to override Provincial Medical data for.	Organization
Organization Classification	The organization classification. This is required even if the organization exists and is already defined as a department.	OrgUnitClassification
Organization Extra Information	Provincial Medical account details are held within flexfield segments and are loaded by the OrgInformation record.	OrgInformation

Organization Attributes

Supply one Organization record for each department you are providing Provincial Medical account override details for.

The Organization record type requires these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the department override.
Name	The name of the department override.
ClassificationName	The name of the organization classification. Specify 'DEPARTMENT'

Organization Classification Attributes

Supply one OrgUnitClassification record for each department you are providing Provincial Medical override details for. The OrgUnitClassification record type requires these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the department classification override.
OrganizationName	The name of the department to override. This must match the value supplied to the Name attribute on the parent Organization record.
ClassificationName	The name of the department classification. Specify 'DEPARTMENT'.

Organization Extra Information Attributes

Supply one OrgInformation record for each department to provide overrides for. These are held in the PER_ORGANIZATION_INFORMATION_EFF flexfield.

The attributes to supply are different for Provincial Medical and Workers Compensation overrides.

Provincial Medical:

Provincial Medical account overrides at the department level are held on the on DEPARTMENT flexfield category with this flexfield context:

- Canada Provincial Medical Department Information (ORA_HRX_CA_PM_DEPT)

Supply these attributes:

HCM Data Loader Attribute	Functional Description
EffectiveStartDate	The start date of the department override information.
OrganizationName	The name of the department. This must match the value supplied to the Name attribute on the parent Organization record.
ClassificationName	The name of the organization classification. Specify 'DEPARTMENT'.

HCM Data Loader Attribute	Functional Description
SequenceNumber	The sequence number for the classification component record. Enter a unique number for each record of this type. If multiple rows for the same type, increase each row number by 1. **Note:** This is sequential according to the component context. For example, if loading 2 provincial medical records and 1 for workers compensation, populate 1 and 2 for the provincial medical records and 1 for the workers compensation record.
EFF_CATEGORY_CODE	The flexfield context category code that is used to supply the Workers Compensation overrides. Specify 'DEPARTMENT'.
FLEX:PER_ORGANIZATION_INFORMATION_EFF	The flexfield context code that is used to supply the Workers Compensation overrides. Specify 'ORA_HRX_CA_PM_DEPT'.
OrgInformationContext	Supply the same value as the FLEX:PER_ORGANIZATION_INFORMATION_EFF attribute value.
oraHrxCaPmDeptUniqueSeg(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PM_DEPT)	The sequence number for the component record. Enter a unique number for each record of this type. If multiple rows for the same type, increase each row number by 1
oraHrxCaPmDeptPsu(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PM_DEPT)	The ID used to identify the Payroll Statutory Unit.
oraHrxCaPmDeptProv(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PM_DEPT)	The geocode that identifies the Provincial Medical Province override value.
oraHrxCaPmDeptAccount(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PM_DEPT)	The Provincial Medical Account Workers Compensation Classification Unit override value of the component record.

Note: Refer to the Cloud Customer Connect topic: *BI Publisher Report: Canada Provincial Medical Accounts for Overrides* for a report that extracts the values required for the PSU and account flexfield segment attributes.

Example of Loading Provincial Medial Account Overrides against a Department for Canada

This Organization.dat file will update the existing Administration department to define the Provincial Medical Account override.

```
METADATA|Organization|EffectiveStartDate|Name|ClassificationName
MERGE|Organization|2016/01/01|Administration|DEPARTMENT

METADATA|OrgUnitClassification|EffectiveStartDate|OrganizationName|ClassificationName
MERGE|OrgUnitClassification|2016/01/01|Administration|DEPARTMENT

METADATA|OrgInformation|EffectiveStartDate|OrganizationName|ClassificationName|
EFF_CATEGORY_CODE|FLEX:PER_ORGANIZATION_INFORMATION_EFF|OrgInformationContext|
SequenceNumber|oraHrxCaPmDeptUniqueSeg(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PM_DEPT)|
oraHrxCaPmDeptPsu(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PM_DEPT)|
oraHrxCaPmDeptProv(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PM_DEPT)|
oraHrxCaPmDeptAccount(PER_ORGANIZATION_INFORMATION_EFF=ORA_HRX_CA_PM_DEPT)
```

```
MERGE|OrgInformation|2018/01/01|Administration|DEPARTMENT|DEPARTMENT|ORA_HRX_CA_PM_DEPT|ORA_HRX_CA_PM_DEPT|
1|1|300100048140041|5|MB8734834
```

Reporting Information

Overview of Reporting Information Cards for Canada

The Reporting Information calculation card for Canada houses critical details.

The Reporting Information calculation card for Canada captures the following categories of data:

- Record of Employment
- Quebec Labour Standards
- Year-end

Record of Employment Data

The Record of Employment (ROE) is a report generated by employers when an employee has an interruption of earnings. Service Canada uses this report to determine Employment Insurance (EI) benefits. The Reporting Information calculation card captures details reported on the ROE including:

- ROE data
- Vacation pay
- Statutory holiday pay
- Other monies
- Special payments
- Comments

Quebec Labour Standards Data

The Quebec Labour Standards report is used to calculate and report employer liability for Quebec Labour Standards. Employers with employees working in Quebec are subject to the levy. If you have a Parity Committee, you must create the Reporting Information card's Quebec Labour Standards Contribution component to capture the name in the following field:

- Parity Committee Name

Year-End Data

RL-2 Slip Information

If the amount entered in the RL-2 Additional Information Box C includes a single payment made under a registered pension plan (RPP) or a deferred profit-sharing plan (DPSP), employers report C-10 followed by the date of the payment. Use the Reporting Information card's Year-End Data component to capture this date in the following field:

- RL-2 RPP or DPSP Single Payment Date

Use the Reporting Information card's Year-End Data component to capture the fields below. These fields are available to override the source of income for an employee. Enter the RL-2 Type of Income. If none of the options apply, select 'Other plan or fund' and enter the RL-2 Type of Plan field. This will report 'AUTRE' in the Provenance box along with Other Info code 201.

- RL-2 Type of Income
- RL-2 Type of Plan

Considerations and Prerequisites

When creating Reporting Information cards, you specify the surrogate ID of the employee's payroll as a context to the card component. Refer to the Cloud Customer Connect topic Report – *Canadian Payrolls* for a report to extract this information.

You can review the Reporting Information cards on the Manage Calculation Cards page.

Reporting Information Card Record Types

The Reporting Information card is bulk-loaded using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various localization requirements.

The Reporting Information Card uses these Calculation Card record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee payroll relationship that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following topics describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Supply a component detail record for each flexfield context required by each card component.	ComponentDetail

Reporting Information Card Hierarchy

Related Topics

- Guidelines for Loading Canadian Reporting Information Cards
- Guidelines for Loading Card Components for Canadian Reporting Information Cards
- Example of Loading ROE Data for Canadian Reporting Information Cards

Guidelines for Loading Canadian Reporting Information Cards

Create Reporting Information Calculation Cards to capture reporting information for employees.

> **Note:** Even if you are updating an existing calculation card and the calculation card itself is not being updated, still include the calculation card record to group other related data supplied in the file.

Reporting Information Calculation Card Attributes

The Reporting Information calculation card for employees uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Reporting Information calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source application owner used to generate the source application ID.
EffectiveStartDate	N/A	The start date of the Reporting Information card. Format YYYY/MM/DD.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Reporting Information'
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.

Supply these attributes against the CalculationCard file discriminator. For example:

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|EffectiveStartDate|LegislativeDataGroupName|
DirCardDefinitionName|AssignmentNumber
MERGE|CalculationCard|VISION|RI8182267|2022/01/31|CA LDG|Reporting Information|E8182267
```

Related Topics

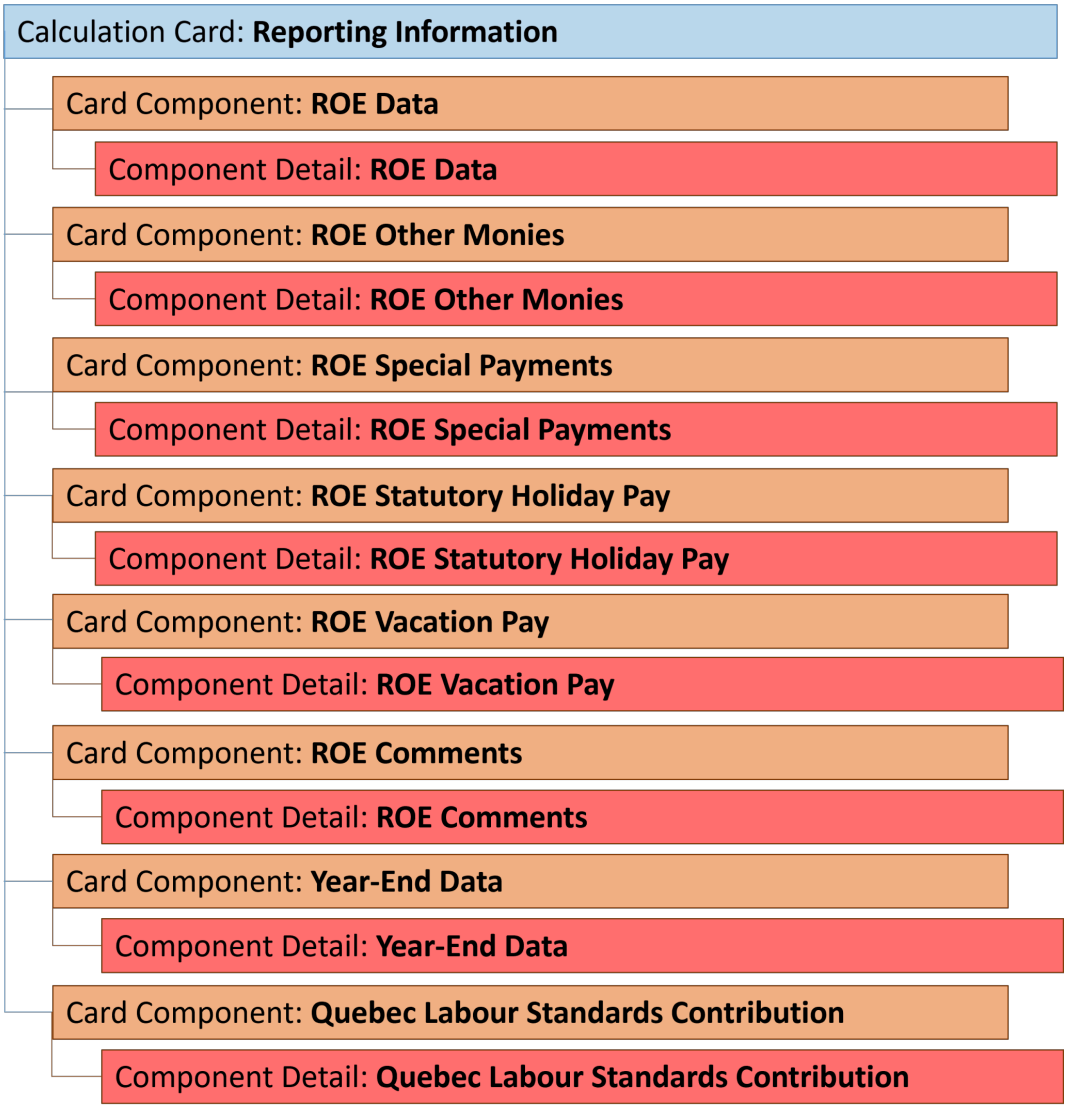
- Overview of Reporting Information Cards for Canada
- Guidelines for Loading Card Components for Canadian Reporting Information Cards

Guidelines for Loading Card Components for Canadian Reporting Information Cards

There are multiple reporting cards available on the Canadian Reporting Information card.

They are loaded using the same record types, the difference being the flexfield context and segments applicable to each card component.

Reporting Information Card Component Record Hierarchy



To create a card component for the Reporting Information card,cardcomponentforou need to supply two record types: Card Component and Component Detail. The detail creates data within flexfield segments applicable to the component.

Card Component Attributes for Reporting Information

All Reporting Information card components uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the **Reporting Information** card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent **Reporting Information** calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the **Reporting Information** card component.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Specify one of: • ROE Data • ROE Comments • ROE Other Monies • ROE Special Payments • ROE Statutory Holiday Pay • ROE Vacation Pay • ROE Comments • Quebec Labour Standards Contribution • Year-end Data
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Specify '1' if using user keys. Not required when source keys are used.
Context1	N/A	The payroll ID of the employee's payroll. Refer to Cloud Customer Connect topic Report: Canadian Payrolls.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
Context2	N/A	The reference code for the card component. A code that uniquely identifies the component. It is user-defined and is used to further specify the component.
Context3	N/A	The name of the employee's tax reporting unit to associate with the component.

These attributes are supplied against the CardComponent file discriminator and must be supplied along with a CalculationCard record for the parent **Reporting Information** card.

Component Detail Attributes for Reporting Information

The component detail record type allows you to upload data into flexfield segments.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context and card component:

Card Component	Flexfield Context Code
ROE Data	ORA_HRX_CA_ROE_REASON
ROE Other Monies	ORA_HRX_CA_ROE_OTHER_MONIES
ROE Special Payments	ORA_HRX_CA_ROE_SPECIAL_PAYMENTS
ROE Statutory Holiday Pay	ORA_HRX_CA_ROE_STATUTORY_HOLIDAY_PAY
ROE Vacation Pay	ORA_HRX_CA_ROE_VACATION_PAY
ROE Comments	ORA_HRX_CA_ROE_COMMENTS
Quebec Labour Standards Contribution	ORA_HRX_CA_LABOUR_STANDARDS
Year-End Data	ORA_HRX_CA_EOY_DATA

Core attributes for Component Details:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName,	The parent card component should be referenced using the same key type used to identify the parent record.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, DirCardCompDefName	When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component. Valid values are: • ROE Data • ROE Other Monies • ROE Special Payments • ROE Statutory Holiday Pay • ROE Vacation Pay • ROE Comments • Quebec Labour Standards Contribution • Year-End Components
DirInformationCategory	N/A	The code for the flexfield context applicable to the card component. These are listed above this table. For example, supply 'ORA_HRX_CA_ROE_REASON' for the ROE Data card component.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parentcard component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.

The attributes used to populate the flexfield segments differ depending on the flexfield context.

You can find the current flexfield segment attribute names for each flexfield context using the View Business Objects task. However, the relevant attributes are listed here.

ROE Data Card Component Flexfield Segment Attributes

These attributes are available for the ROE Data card component and ORA_HRX_CA_ROE_REASON flexfield context code:

HCM Data Loader Attribute	UI Prompt	Functional Description
oraHrxCaRoeReasonCode(Deduction Developer DF=ORA_HRX_CA_ROE_REASON)	Reason for Issuing ROE	The reason for issuing the Record of Employment. This attribute is validated using the lookup type ORA_HRX_CA_ROE_REASON_CODES. This attribute is mandatory when loading the ROE Data card component.
expectedRecallCode(Deduction Developer DF=ORA_HRX_CA_ROE_REASON)	Expected Recall Code	The expected recall code for the employee. This attribute is validated using the lookup ORA_HRX_CA_ROE_RECALL_CODES. Specify the lookup code to this attribute. Enter 'ORA_U' if the value is unknown.
expectedRecallDate(Deduction Developer DF=ORA_HRX_CA_ROE_REASON)	Expected Recall Date	The expected recall date for the employee
overrideFirstDayWorked(Deduction Developer DF=ORA_HRX_CA_ROE_REASON)	Override First Day Worked	The override of the first day worked for the employee.
overrideTerminationDate(Deduction Developer DF=ORA_HRX_CA_ROE_REASON)	Override Last Day for Which Paid	The override of the last day for which the employee was paid.

ROE Other Monies Card Component Flexfield Segment Attributes

These attributes are available for the ROE Other Monies card component and ORA_HRX_CA_ROE_OTHER_MONIES flexfield context code:

HCM Data Loader Attribute	UI Prompt	Functional Description
payType(Deduction Developer DF=ORA_HRX_CA_ROE_OTHER_MONIES)	Pay Type	The pay type associated with the ROE Other Monies. This attribute is mandatory and validated using the lookup ORA_HRX_CA_ROE_PAY_TYPE. Supply the lookup code.
payCode(Deduction Developer DF=ORA_HRX_CA_ROE_OTHER_MONIES)	Pay Code	The pay code associated with the ROE Other Monies. This attribute is mandatory and validated using the lookup ORA_HRX_CA_ROE_OM_PAY_CODES. Supply the lookup code.
startDate(Deduction Developer DF=ORA_HRX_CA_ROE_OTHER_MONIES)	Start Date	The start date associated with the ROE Other Monies.
endDate(Deduction Developer DF=ORA_HRX_CA_ROE_OTHER_MONIES)	End Date	The end date associated with the ROE Other Monies.
amount(Deduction Developer DF=ORA_HRX_CA_ROE_OTHER_MONIES)	Amount	The amount associated with the ROE Other Monies.
otherMoniesAmountType(Deduction Developer DF=ORA_HRX_ CA_ROE_OTHER_MONIES)	Report the amount in the final pay period?	Specifies whether the ROE Other Monies amount is reported in the final pay period or not. Specify 'Y' or 'N'. The default is 'Y'.

ROE Special Payments Card Component Flexfield Segment Attributes

These attributes are available for the ROE Special Payments card component and ORA_HRX_CA_ROE_SPECIAL_PAYMENTS flexfield context code:

HCM Data Loader Attribute	UI Prompt	Functional Description
payCode(Deduction Developer DF=ORA_HRX_CA_ROE_SPECIAL_PAYMENTS)	Pay Code	The pay code associated with the ROE Special Payments. This attribute is mandatory and validated using the lookup ORA_HRX_CA_ROE_SP_PAY_CODES
startDate(Deduction Developer DF=ORA_HRX_CA_ROE_SPECIAL_PAYMENTS)	Start Date	The start date associated with the ROE Special Payments.
endDate(Deduction Developer DF=ORA_HRX_CA_ROE_SPECIAL_PAYMENTS)	End Date	The end date associated with the ROE Special Payments.
amount(Deduction Developer DF=ORA_HRX_CA_ROE_SPECIAL_PAYMENTS)	Amount	The amount associated with the ROE Special Payments.
period(Deduction Developer DF=ORA_HRX_CA_ROE_SPECIAL_PAYMENTS)	Period	The period associated with the ROE Special Payments. This attribute is validated using the lookup ORA_HRX_CA_ROE_SP_PERIOD. Supply the lookup code.

ROE Statutory Holiday Pay Card Component Flexfield Segment Attributes

These attributes are available for the ROE Statutory Holiday Pay card component and ORA_HRX_CA_ROE_STATUTORY_HOLIDAY_PAY flexfield context code:

HCM Data Loader Attribute	UI Prompt	Functional Description
payDate(Deduction Developer DF=ORA_HRX_CA_ROE_STATUTORY_HOLIDAY_PAY)	Date	The pay date associated with the ROE Statutory Holiday Pay. This is mandatory when creating ROE Statutory Holiday Pay card components.
amount(Deduction Developer DF=ORA_HRX_CA_ROE_STATUTORY_HOLIDAY_PAY)	Amount	The amount associated with the ROE Statutory Holiday Pay. This is mandatory when creating ROE Statutory Holiday Pay card components.

ROE Vacation Pay Card Component Flexfield Segment Attributes

These attributes are available for the ROE Vacation Pay card component and ORA_HRX_CA_ROE_VACATION_PAY flexfield context code:

HCM Data Loader Attribute	UI Prompt	Functional Description
payType(Deduction Developer DF=ORA_HRX_CA_ROE_VACATION_PAY)	Pay Type	The pay type associated with the ROE Vacation Pay. This attribute is mandatory and validated using the lookup ORA_HRX_CA_ROE_PAY_TYPE. Supply the lookup code.
payCode(Deduction Developer DF=ORA_HRX_CA_ROE_VACATION_PAY)	Pay Code	The pay code associated with the ROE Vacation Pay. This attribute is mandatory and validated using the lookup ORA_HRX_CA_ROE_VP_PAY_CODES. Supply the lookup code.
startDate(Deduction Developer DF=ORA_HRX_CA_ROE_VACATION_PAY)	Start Date	The start date associated with the ROE Vacation Pay.
endDate(Deduction Developer DF=ORA_HRX_CA_ROE_VACATION_PAY)	End Date	The end date associated with the ROE Vacation Pay.

HCM Data Loader Attribute	UI Prompt	Functional Description
amount(Deduction Developer DF=ORA_HRX_CA_ROE_VACATION_PAY)	Amount	The amount associated with the ROE Vacation Pay.

ROE Comments Card Component Flexfield Segment Attributes

These attributes are available for the ROE Comments card component and ORA_HRX_CA_ROE_COMMENTS flexfield context code:

HCM Data Loader Attribute	UI Prompt	Functional Description
comments(Deduction Developer DF=ORA_HRX_CA_ROE_COMMENTS)	Comments	Comments entered as free-form text for the ROE. The maximum value for entry is 150 characters.

Quebec Labour Standards Contribution Card Component Flexfield Segment Attributes

These attributes are available for the Quebec Labour Standards Contribution card component and ORA_HRX_CA_LABOUR_STANDARDS flexfield context code:

HCM Data Loader Attribute	UI Prompt	Functional Description
parityCommitteeName(Deduction Developer DF=ORA_HRX_CA_LABOUR_STANDARDS)	Parity Committee Name	The name of the parity committee. This attribute is validated using the lookup ORA_HRX_CA_PARITY_COMMITTEE_NAME.

Year-End Data Card Component Flexfield Segment Attributes

These attributes are available for the Year-End Data card component and ORA_HRX_CA_EOY_DATA flexfield context code:

HCM Data Loader Attribute	UI Prompt	Functional Description
RL-2RPPorDPSPSinglePaymentDate(Deduction Developer DF=ORA_HRX_CA_EOY_DATA)	RL-2 RPP or DPSP Single Payment Date	Date of the payment made under a registered pension plan (RPP) or a deferred profit-sharing plan (DPSP).
rl2IncomeType(Deduction Developer DF=ORA_HRX_CA_EOY_DATA)	RL-2 Type of Income	An override to the source of income for an employee. If none of the options apply, select 'Other plan or fund' and enter the RL-2 Type of Plan field (see below). This will report 'AUTRE' in the Provenance box along with Other Info code 201.
rl2PlanType(Deduction Developer DF=ORA_HRX_CA_EOY_DATA)	RL-2 Type of Plan	The type of plan to report on the RL-2, if you selected 'Other plan or fund' for the RL-2 Type of Income.

Related Topics

- Overview of Reporting Information Cards for Canada
- Guidelines for Loading Canadian Reporting Information Cards
- Example of Loading ROE Data for Canadian Reporting Information Cards

Example of Loading ROE Data for Canadian Reporting Information Cards

In this example a Reporting Information card is created for reporting ROE information.

The ROE flexfield segments populated are:

Flexfield Segment	Attribute Value	Meaning
Reason for Issuing ROE	ORA_D00	Illness of injury
Expected Recall Code	ORA_Y	Expected data of recall
Expected Recall Date	2022/01/12	N/A
Override Last Day for Which Paid	2022/01/05	N/A

Use the CalculationCard.dat file name to load Reporting Information card data.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|EffectiveStartDate|LegislativeDataGroupName|
DirCardDefinitionName|AssignmentNumber
MERGE|CalculationCard|VISION|RI8182267|2022/01/31|CA LDG|Reporting Information|E8182267

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|EffectiveStartDate|LegislativeDataGroupName|
DirCardId(SourceSystemId)|DirCardCompDefName|Context1|Context2
MERGE|CardComponent|VISION|RI8182267_DATA|2022/01/31|CA LDG|RI8182267|ROE Data|300100038120600|

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|EffectiveStartDate|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|DirCardCompDefName|DirInformationCategory|FLEX:Deduction Developer
 DF|oraHrxCaRoeReasonCode(Deduction Developer DF=ORA_HRX_CA_ROE_REASON)|expectedRecallCode(Deduction
 Developer DF=ORA_HRX_CA_ROE_REASON)|expectedRecallDate(Deduction Developer DF=ORA_HRX_CA_ROE_REASON)|
overrideTerminationDate(Deduction Developer DF=ORA_HRX_CA_ROE_REASON)
MERGE|ComponentDetail|VISION|RI8182267_DATA_FLX|2022/01/31|CA LDG|RI8182267_DATA|ROE Data|
ORA_HRX_CA_ROE_REASON|ORA_HRX_CA_ROE_REASON|ORA_D00|ORA_Y|2022/01/12|2022/01/05
```

Related Topics

- Overview of Reporting Information Cards for Canada
- Guidelines for Loading Canadian Reporting Information Cards
- Guidelines for Loading Card Components for Canadian Reporting Information Cards

Example of Loading ROE Other Monies for Canadian Reporting Information Cards for Canada

In this example a Reporting Information card is created with ROE Other Monies.

The ROE Other Monies flexfield segments populated are:

Flexfield Segment	Attribute Value	Meaning
Pay Type	ORA_A	Amount
Pay Code	ORA_B11	Other bonus
Pay Code	2022/01/01	N/A
End Date	2022/12/31	N/A
Amount	250	N/A
Report the amount in the final pay period?	Y	Yes

Use the CalculationCard.dat file name to load Reporting Information card data.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|EffectiveStartDate|LegislativeDataGroupName|
DirCardDefinitionName|AssignmentNumber
MERGE|CalculationCard|VISION|RI8182267|2022/01/31|CA LDG|Reporting Information|E8182267

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|EffectiveStartDate|LegislativeDataGroupName|
DirCardId(SourceSystemId)|DirCardCompDefName|Context1|Context2
MERGE|CardComponent|VISION|RI8182267_MNY|2022/01/31|CA LDG|RI8182267|ROE Data|300100038120600|Bonus 2022

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|EffectiveStartDate|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|DirCardCompDefName|DirInformationCategory|FLEX:Deduction Developer
 DF|payType(Deduction Developer DF=ORA_HRX_CA_ROE_OTHER_MONIES)|payCode(Deduction Developer
 DF=ORA_HRX_CA_ROE_OTHER_MONIES)|startDate(Deduction Developer DF=ORA_HRX_CA_ROE_OTHER_MONIES)|
 endDate(Deduction Developer DF=ORA_HRX_CA_ROE_OTHER_MONIES)|amount(Deduction Developer
 DF=ORA_HRX_CA_ROE_OTHER_MONIES)|otherMoniesAmountType(Deduction Developer DF=ORA_HRX_CA_ROE_OTHER_MONIES)
MERGE|ComponentDetail|VISION|RI8182267_MNY_FLX|2022/01/31|CA LDG|RI8182267_MNY|ROE Data|
ORA_HRX_CA_ROE_OTHER_MONIES|ORA_HRX_CA_ROE_OTHER_MONIES|ORA_A||2022/01/01|2022/12/31|250|Y
```

Related Topics

- Overview of Reporting Information Cards for Canada
- Guidelines for Loading Canadian Reporting Information Cards
- Guidelines for Loading Card Components for Canadian Reporting Information Cards

Adjusting Balances

Overview of Loading Balance Adjustments for Canada

Consider the following points when performing balance adjustments for Canada.

Perform balance adjustments to:

- Correct the entries from the Load Initial Balances process that were uploaded during conversion.
- Correct balances that were loaded:
  - With incorrect tax balance dimensions
  - With incorrect province tax jurisdiction
  - With over or understated taxable limits
- Correct balances that were not loaded at all and a payroll was process for the employee.
- Load balances to Gross-to-Net Calculations done by a third-party, but the employer is still responsible to report, such as Disability Pay, Stock Options, Moving and Relocation, and so on.
- Zero out arrears upon termination

Considerations for Adjusting Balances for Canada

In addition to the general guidance provided in the Overview of Loading Balance Adjustments topic, consider the following when adjusting balances for Canada:

- You must process terminated employees in a separate batch with a process date prior to the date they were terminated.
- If an employee has transferred between provinces, you must create a separate adjustment line item for each province. Adjustments use the province as a context.
- Fields capturing geography codes to indicate the province require the actual geocode. Refer to the *Geography Codes for Canadian Provinces* topic.
- If an employee has transferred from one tax reporting unit (TRU) to another within the calendar year, and you need to perform a balance adjustment for either TRU, then you must perform a separate adjustment for each, using a process date as of when the transfer took place. When defining the adjustment in the batch, you must enter the Calculation Breakdown Component value. This value is the same as the TRU you are currently adjusting. There is a 1-to-1 relationship between the TRU and the Calculation Breakdown Component. For example, if the TRU is CA MAIN TRU, then the Calculation Breakdown Component must also be CA MAIN TRU.

Contexts for Adjusting Balances for Canada

For the purposes of balance adjustment, some of the dimensions require require different contexts for the components of the dimensions.

Statutory Report Type

The Statutory Report Type input value defines the associated Reporting Type for end-of-year reporting for earnings and deductions. The report types are predefined for all seeded secondary classifications for Canada, and if you create a secondary classification, this value is mandatory. This context is required for some dimensions.

Values for the Statutory Reporting Type context are:

- T4_RL1 (Meaning = T4 and RL-1)
- T4A_RL1 (Meaning = T4A and RL-1)
- T4A_RL2 (Meaning = T4A and RL-2)

Reporting Time Period

The Reporting Time Period input value defines a unique value for each pay period, automatically generated by the payroll run. The context would be used in a scenario where a payroll transfer occurs. It only applies to CPP and QPP. The Reporting Time Period value is derived by:

- Current pay period number concatenated with pay periods in the year
  - For example, the Reporting Time Period for a monthly payroll processed in February is “212”
  - 2 = Current pay period of February + 12 = 12 monthly payrolls in the year

Balance Adjustment Elements

When performing adjustments on elements, the process uses one of the following to adjust the balance:

- Primary, or base, element
- Results element

The Results element is a secondary element created when you define the primary element.

For adjustments on predefined tax elements, the process uses the primary element to adjust the balance.
This table outlines which element and input value to use to adjust balances for each primary and secondary classification.

Primary Classification	Secondary Classification	Element Generated from the Template	Input Value
Standard Earnings	All	<User element> Results	Earnings
Supplemental Earnings	All	<User element> Results	Earnings
Taxable Benefits	All	<User element> Results	Earnings
Nonpayroll Payment	All	<User element> Results	Earnings
Pretax Deductions	All	<User element> Results	Pay Value
Employee Tax Deductions	All	<Seeded element>	<Tax> Withheld
Employer Liabilities	All	<User element>	Pay Value
Involuntary Deductions	All	<User element> Results	DeductionsCalculated
Involuntary Deductions	All (Fees)	<User element> Organization Fee Results	FeeCalculated
Involuntary Deductions	All (Fees)	<User element> Person Fee Results	FeeCalculated
Involuntary Deductions	All (Fees)	<User element> Processing Fee Results	FeeCalculated
Voluntary Deductions	All	<User element> Results	Pay Value

Related Topics

- Overview of Loading Balance Adjustments
- Steps to Adjust Balances
- Balance Adjustment Record Types
- Geography Codes for Canadian Provinces

Example of Loading Federal Tax Withheld Balance Adjustments for Canada

This example creates a batch header and group record, with various lines and values to adjust the Federal Tax Withheld balance for two employees to increase the withheld balances by $10.

> **Note:** When adjusting balances you may also need to adjust any related balances, depending on your business needs. This is simply an example to illustrate the process.

Load the Balance Adjustment Header and Group

Use the BalanceAdjustmentHeader.dat file to create the batch header and group records.

```
METADATA|BalanceAdjustmentHeader|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|CA LDG|Adjust Federal Taxes

METADATA|BalanceAdjustmentGroup|LegislativeDataGroupName|BatchName|EffectiveDate|PayrollName|
ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|CA LDG|Adjust Federal Taxes|2020/01/31|CA Semimonthly|CA Semimonthly
 Consolidation Set|N|N
```

The lines and values file can be simplified if Source Keys are used, as the user keys include a large number of attributes. However, this requires that source keys are used to identify the header and group. For example:

```
METADATA|BalanceAdjustmentHeader|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|VISION|ADJ_FED_TAX_2018|CA LDG|Adjust Federal Taxes 2018

METADATA|BalanceAdjustmentGroup|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
BalAdjBatchId(SourceSystemId)|EffectiveDate|PayrollName|ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|VISION|ADJ_FED_TAX_2018_GRP|CA LDG|ADJ_FED_TAX_2018|2018/01/31|CA Semimonthly|
CA Semimonthly Consolidation Set|N|N
```

Load the Balance Adjustment Lines and Values

The Balance Adjustment Lines files lines specify the element you will supply values for:

Batch Line Sequence	Assignment Number	Element Name	Tax Reporting Unit Name
1	E182111	Federal Taxes	CA Tax Reporting Unit
2	E182110	Federal Taxes	CA Tax Reporting Unit

The Balance Adjustment Value file lines specify the input values for the element named in the line record. The Entry Value is the value for the Input Value Name of the Element Name. For example, in the first line below, the Entry Value of $10 is the value specified for the Tax Calculated input value for the Federal Taxes element.

You must also specify the contexts for the Province and Reporting Type. That is the reason there are 3 value records for the 1 line record above for the Federal Taxes element. In the value record below where Province is the Input Value Name, the Entry Value corresponds to the geography code of the province. For example, the value '1' corresponds to the province of Alberta.

In the value record below where Reporting Type is the Input Value Name, the Entry Value corresponds to the statutory reporting code.

For example, the value 'T4_RL1' corresponds to 'T4 and RL-1' for the end-of-year reporting type of the element.

Batch Line Sequence	Element Name	Input Value Name	Entry Value
1	Federal Taxes	Federal Withheld	10
1	Federal Taxes	Province	1
1	Federal Taxes	Reporting Type	T4_RL1
2	Federal Taxes	Federal Withheld	10
2	Federal Taxes	Province	21
2	Federal Taxes	Reporting Type	T4_RL1

Use the BalanceAdjustmentLine.dat file to create the batch line and value records.

```
METADATA|BalanceAdjustmentLine|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|AssignmentNumber|ElementName|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|CA LDG|Adjust Federal Taxes|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/01/31|1|E182111|Federal Taxes|CA Tax Reporting Unit
MERGE|BalanceAdjustmentLine|CA LDG|Adjust Federal Taxes|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/01/31|2|E182110|Federal Taxes|CA Tax Reporting Unit

METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|InputValueName|ElementName|EntryValue
MERGE|BalanceAdjustmentValue|CA LDG|Adjust Federal Taxes|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/01/31|1|Federal Withheld|Federal Taxes|10
MERGE|BalanceAdjustmentValue|CA LDG|Adjust Federal Taxes|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/01/31|1|Province|Federal Taxes|1
MERGE|BalanceAdjustmentValue|CA LDG|Adjust Federal Taxes|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/01/31|1|Reporting Type|Federal Taxes|T4_RL1
MERGE|BalanceAdjustmentValue|CA LDG|Adjust Federal Taxes|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/01/31|2|Federal Withheld|Federal Taxes|10
MERGE|BalanceAdjustmentValue|CA LDG|Adjust Federal Taxes|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/01/31|2|Province|Federal Taxes|21
MERGE|BalanceAdjustmentValue|CA LDG|Adjust Federal Taxes|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/01/31|2|Reporting Type|Federal Taxes|T4_RL1
```

When source keys are supplied the balance adjustment group can be identified by the two source key attributes as opposed to the five user attributes. Similarly, the balance adjustment line can be identified by the two source key attributes as opposed to the six user key attributes.

```
METADATA|BalanceAdjustmentLine|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
BalAdjGroupId(SourceSystemId)|BatchLineSequence|AssignmentNumber|ElementName|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|VISION|ADJ_FED_TAX_2018_GRP_1|CA LDG|ADJ_FED_TAX_2018_GRP|1|E182111|Federal
 Taxes|CA Tax Reporting Unit
```

```
MERGE|BalanceAdjustmentLine|VISION|ADJ_FED_TAX_2018_GRP_2|CA LDG|ADJ_FED_TAX_2018_GRP|2|E182110|Federal
 Taxes|CA Tax Reporting Unit

METADATA|BalanceAdjustmentValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
BatchNameBalAdjLineId(SourceSystemId)|InputValueName|EntryValue|ElementName
MERGE|BalanceAdjustmentValue|VISION|ADJ_FED_TAX_2018_GRP_1_FW|CA LDG|ADJ_FED_TAX_2018_GRP_1|Federal
 Withheld|10|Federal Taxes
MERGE|BalanceAdjustmentValue|VISION|ADJ_FED_TAX_2018_GRP_1_PR|CA LDG|ADJ_FED_TAX_2018_GRP_1|Province|1|
Federal Taxes
MERGE|BalanceAdjustmentValue|VISION|ADJ_FED_TAX_2018_GRP_1_RT|CA LDG|ADJ_FED_TAX_2018_GRP_1|Reporting Type|
T4_RL1|Federal Taxes
MERGE|BalanceAdjustmentValue|VISION|ADJ_FED_TAX_2018_GRP_2_FW|CA LDG|ADJ_FED_TAX_2018_GRP_2|Federal
 Withheld|10|Federal Taxes
MERGE|BalanceAdjustmentValue|VISION|ADJ_FED_TAX_2018_GRP_2_PR|CA LDG|ADJ_FED_TAX_2018_GRP_2|Province|21|
Federal Taxes
MERGE|BalanceAdjustmentValue|VISION|ADJ_FED_TAX_2018_GRP_2_RT|CA LDG|ADJ_FED_TAX_2018_GRP_2|Reporting Type|
T4_RL1|Federal Taxes
```

Example of Loading Quebec Labour Standard Balances Balance Adjustments: Exempt CCQ Employee Wages – With Excess Remuneration

At year-end, if any wages are adjusted, you must also adjust the related Quebec Labour Standards balances.

This example creates a batch header and group record, with various lines and values to adjust some Quebec Labour Standards balances for one employee. In this case, the CCQ Employee Wages is adjusted by $5,000. Other related QLS balances are adjusted as required. In this example, the annual maximum limit is $78,500. No adjustments are required to Gross Earnings.

Balance Name	CCQ Employee Wages
YTD Balance Prior to Adjusting	0
Adjustment Value	5,000
YTD Balance After Adjusting	5,000
Notes	Adjusted amount to $5,000. This is an exempt balance and reduces subject.

Balance Name	Quebec Labour Standards Gross
YTD Balance Prior to Adjusting	85,000
Adjustment Value	N/A
YTD Balance After Adjusting	85,000
Notes	No changes required because Quebec Labour Standards Exempt was adjusted.

Balance Name	Quebec Labour Standards Subject
YTD Balance Prior to Adjusting	85,000

Adjustment Value	-5,000
YTD Balance After Adjusting	80,000
Notes	Adjusted to reduce for exempt wages.

Balance Name	Quebec Labour Standards Exempt
YTD Balance Prior to Adjusting	0
Adjustment Value	5,000
YTD Balance After Adjusting	5,000
Notes	Adjusted to include as exempt wages.

Balance Name	Quebec Labour Standards Excess
YTD Balance Prior to Adjusting	6,500
Adjustment Value	-5,000
YTD Balance After Adjusting	1,500
Notes	Adjustment to reduce the same as subject amount.

Balance Name	Quebec Labour Standards Taxable
YTD Balance Prior to Adjusting	78,500
Adjustment Value	N/A
YTD Balance After Adjusting	78,500
Notes	Taxable is capped at 2020 annual maximum limit.

Balance Name	Quebec Labour Standards Liability
YTD Balance Prior to Adjusting	54.95
Adjustment Value	N/A
YTD Balance After Adjusting	54.95
Notes	Liability is capped at 2020 annual maximum limit.

Note: When adjusting balances you may also need to adjust any related balances, depending on your business needs. This is simply an example to illustrate the process.

Load the Balance Adjustment Header and Group

Use the BalanceAdjustmentHeader.dat file to create the batch header and group records.

```
METADATA|BalanceAdjustmentHeader|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|CA LDG|EX_CCQ_EXCESS_RENUMERATION

METADATA|BalanceAdjustmentGroup|LegislativeDataGroupName|BatchName|EffectiveDate|PayrollName|
ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|CA LDG|EX_CCQ_EXCESS_RENUMERATION|2020/12/31|CA Semimonthly|CA Semimonthly
 Consolidation Set|N|N
```

Load the Balance Adjustment Lines and Values

The Balance Adjustment Lines files lines specify the element you will supply values for:

Batch Line Sequence	Assignment Number	Element Name	Tax Reporting Unit Name
1	E192278	Quebec Labour Standards Results	CA Tax Reporting Unit

The Balance Adjustment Value file lines specify the input values for the element named in the line record. The Entry Value is the value for the Input Value Name of the Element Name. For example in the third line below, the Entry Value of -5,000 is the value specified for the Quebec Labour Standards Subject input value for the Quebec Labour Standards Results element.

You must also specify the contexts for the Province, Quebec Identification Number, and Reporting Type.

Batch Line Sequence	Element Name		
1	Quebec Labour Standards Results	Province	19
		Quebec Identification Number	7453627101RS0001
		Reporting Type	T4_RL1
		CCQ Employee Wages	5000
		Quebec Labour Standards Subject	-5000
		Quebec Labour Standards Exempt	5000
		Quebec Labour Standards Excess	-5000

Use the BalanceAdjustmentLine.dat file to create the batch line and value records.

```
METADATA|BalanceAdjustmentLine|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|AssignmentNumber|ElementName|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|CA LDG|EX_CCQ_EXCESS_RENUMERATION|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|E192278|Quebec Labour Standards Results|CA Tax Reporting Unit

METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|InputValueName|ElementName|EntryValue
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_RENUMERATION|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Province|Quebec Labour Standards Results|19
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_RENUMERATION|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Identification Number|Quebec Labour Standards Results|7453627101RS0001
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_RENUMERATION|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Reporting Type|Quebec Labour Standards Results|T4_RL1
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_RENUMERATION|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|CCQ Employee Wages|Quebec Labour Standards Results|5000
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_RENUMERATION|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Subject|Quebec Labour Standards Results|-5000
```

```
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_RENUMERATION|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Exempt|Quebec Labour Standards Results|5000
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_RENUMERATION|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Excess|Quebec Labour Standards Results|-5000
```

Example of Loading Quebec Labour Standards Balance Adjustments: Exempt CCQ Employee Wages – Without Excess Remuneration

This example creates a batch header and group record, with various lines and values to adjust some Quebec Labour Standards balances for one employee - exempt CCQ employee wages – without excess remuneration.

At year-end, if any wages are adjusted, you must also adjust the related Quebec Labour Standards balances.

In this case, the CCQ Employee Wages is adjusted by $10,000. Other QLS balances are adjusted as required. In this example, the annual maximum limit is $78,500. No adjustments are required to Gross Earnings.

Balance Name	CCQ Employee Wages
YTD Balance Prior to Adjusting	0
Adjustment Value	10,000
YTD Balance After Adjusting	10,000
Notes	Adjusted amount to $10,000. This is an exempt balance and reduces subject

Balance Name	Quebec Labour Standards Gross
YTD Balance Prior to Adjusting	85,000
Adjustment Value	N/A
YTD Balance After Adjusting	85,000
Notes	No changes required because Quebec Labour Standards Exempt was adjusted.

Balance Name	Quebec Labour Standards Subject
YTD Balance Prior to Adjusting	85,000
Adjustment Value	-10,000
YTD Balance After Adjusting	75,000
Notes	Adjusted to reduce for exempt wages.

Balance Name	Quebec Labour Standards Exempt
YTD Balance Prior to Adjusting	0

Adjustment Value	10,000
YTD Balance After Adjusting	10,000
Notes	Adjusted to include as exempt wages.

Balance Name	Quebec Labour Standards Excess
YTD Balance Prior to Adjusting	6,500
Adjustment Value	-6,500
YTD Balance After Adjusting	0
Notes	Adjustment to reduce the same as subject amount.

Balance Name	Quebec Labour Standards Taxable
YTD Balance Prior to Adjusting	78,500
Adjustment Value	-3,500
YTD Balance After Adjusting	74,000
Notes	Adjusted to reduce to match taxable.

Balance Name	Quebec Labour Standards Liability
YTD Balance Prior to Adjusting	54.95
Adjustment Value	-2.45
YTD Balance After Adjusting	52.50
Notes	Adjusted to reflect calculated value (75,000 * .0007 = 52.50)

Note: When adjusting balances you may also need to adjust any related balances, depending on your business needs. This is simply an example to illustrate the process.

Load the Balance Adjustment Header and Group

Use the BalanceAdjustmentHeader.dat file to create the batch header and group records.

```
METADATA|BalanceAdjustmentHeader|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION

METADATA|BalanceAdjustmentGroup|LegislativeDataGroupName|BatchName|EffectiveDate|PayrollName|
ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION|2020/12/31|CA Semimonthly|CA Semimonthly
 Consolidation Set|N|N
```

Load the Balance Adjustment Lines and Values

The Balance Adjustment Lines files lines specify the element you will supply values for:

Batch Line Sequence	Assignment Number	Element Name	Tax Reporting Unit Name
1	E192278	Quebec Labour Standards Results	CA Tax Reporting Unit

The Balance Adjustment Value file lines specify the input values for the element named in the line record. The Entry Value is the value for the Input Value Name of the Element Name. For example in the third line below, the Entry Value of -5,000 is the value specified for the Quebec Labour Standards Subject input value for the Quebec Labour Standards Results element.

You must also specify the contexts for the Province, Quebec Identification Number, and Reporting Type.

Batch Line Sequence	Element Name		
1	Quebec Labour Standards Results	Province	19
		Quebec Identification Number	7453627101RS0001
		Reporting Type	T4_RL1
		CCQ Employee Wages	10,000
		Quebec Labour Standards Subject	-10,000
		Quebec Labour Standards Exempt	10,000
		Quebec Labour Standards Excess	-6,500
		Quebec Labour Standards Taxable	-3,500
		Quebec Labour Standards Liability	-2.45

Use the BalanceAdjustmentLine.dat file to create the batch line and value records.

```
METADATA|BalanceAdjustmentLine|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|AssignmentNumber|ElementName|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|E192278|Quebec Labour Standards Results|CA Tax Reporting Unit

METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|InputValueName|ElementName|EntryValue
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION|CA Semimonthly|CA Semimonthly
 Consolidation Set|2020/12/31|1|Province|Quebec Labour Standards Results|19
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION|CA Semimonthly|CA Semimonthly
 Consolidation Set|2020/12/31|1|Quebec Identification Number|Quebec Labour Standards Results|
7453627101RS0001
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION|CA Semimonthly|CA Semimonthly
 Consolidation Set|2020/12/31|1|Reporting Type|Quebec Labour Standards Results|T4_RL1
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION|CA Semimonthly|CA Semimonthly
 Consolidation Set|2020/12/31|1|CCQ Employee Wages|Quebec Labour Standards Results|10000
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION|CA Semimonthly|CA Semimonthly
 Consolidation Set|2020/12/31|1|Quebec Labour Standards Subject|Quebec Labour Standards Results|-10000
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION|CA Semimonthly|CA Semimonthly
 Consolidation Set|2020/12/31|1|Quebec Labour Standards Exempt|Quebec Labour Standards Results|10000
```

```
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION|CA Semimonthly|CA Semimonthly
 Consolidation Set|2020/12/31|1|Quebec Labour Standards Excess|Quebec Labour Standards Results|-6500
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION|CA Semimonthly|CA Semimonthly
 Consolidation Set|2020/12/31|1|Quebec Labour Standards Taxable|Quebec Labour Standards Results|-3500
MERGE|BalanceAdjustmentValue|CA LDG|EX_CCQ_EXCESS_WO_RENUMERATION|CA Semimonthly|CA Semimonthly
 Consolidation Set|2020/12/31|1|Quebec Labour Standards Liability|Quebec Labour Standards Results|-2.45
```

Example of Loading Quebec Labour Standards Balance Adjustments: Indemnities for Damages – With Excess Remuneration

This example creates a batch header and group record, with various lines and values to adjust some Quebec Labour Standards balances for one employee - indemnities for damages – with excess remuneration.

At year-end, if any wages are adjusted, you must also adjust the related Quebec Labour Standards balances. In this case, the Indemnities for Damages balance is adjusted by $20,000. Other QLS balances are adjusted as required. In this example, the annual maximum limit is $78,500. No adjustments are required to Gross Earnings.

Balance Name	Indemnities for Damages
YTD Balance Prior to Adjusting	0
Adjustment Value	20,000
YTD Balance After Adjusting	20,000
Notes	Adjusted amount to $20,000. This adds to QLS gross.

Balance Name	Quebec Labour Standards Gross
YTD Balance Prior to Adjusting	75,000
Adjustment Value	20,000
YTD Balance After Adjusting	95,000
Notes	Adjusted since Indemnities for Damages was adjusted, it adds to QLS gross.

Balance Name	Quebec Labour Standards Subject
YTD Balance Prior to Adjusting	75,000
Adjustment Value	20,000
YTD Balance After Adjusting	95,000
Notes	Adjusted to reflect increase in QLS gross.

Balance Name	Quebec Labour Standards Exempt

YTD Balance Prior to Adjusting	0
Adjustment Value	N/A
YTD Balance After Adjusting	0
Notes	No adjustment necessary, no exempt wages.

Balance Name	Quebec Labour Standards Excess
YTD Balance Prior to Adjusting	0
Adjustment Value	16,500
YTD Balance After Adjusting	16,500
Notes	Adjusted to reflect amount over the annual limit (95,000 – 78,500).

Balance Name	Quebec Labour Standards Taxable
YTD Balance Prior to Adjusting	75,000
Adjustment Value	3,500
YTD Balance After Adjusting	78,500
Notes	Adjusted to reflect subject amount, up to the limit.

Balance Name	Quebec Labour Standards Liability
YTD Balance Prior to Adjusting	52.50
Adjustment Value	2.45
YTD Balance After Adjusting	54.95
Notes	Adjusted to reflect calculated value (78,500 * .0007 = 54.95).

Note: When adjusting balances you may also need to adjust any related balances, depending on your business needs. This is simply an example to illustrate the process.

Load the Balance Adjustment Header and Group

Use the BalanceAdjustmentHeader.dat file to create the batch header and group records.

```
METADATA|BalanceAdjustmentHeader|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|CA LDG|INDEM_DAM_EXCESS_RENUM

METADATA|BalanceAdjustmentGroup|LegislativeDataGroupName|BatchName|EffectiveDate|PayrollName|
ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|CA LDG|INDEM_DAM_EXCESS_RENUM|2020/12/31|CA Semimonthly|CA Semimonthly
 Consolidation Set|N|N
```

Load the Balance Adjustment Lines and Values

The Balance Adjustment Lines files lines specify the element you will supply values for:

Batch Line Sequence	Assignment Number	Element Name	Tax Reporting Unit Name
1	E192278	Quebec Labour Standards Results	CA Tax Reporting Unit

The Balance Adjustment Value file lines specify the input values for the element named in the line record. The Entry Value is the value for the Input Value Name of the Element Name. For example in the fourth line below, the Entry Value of 20,000 is the value specified for the Indemnities for Damages input value for the Quebec Labour Standards Results element.

You must also specify the contexts for the Province, Quebec Identification Number, and Reporting Type.

Batch Line Sequence	Element Name		
1	Quebec Labour Standards Results	Province	19
		Quebec Identification Number	7453627101RS0001
		Reporting Type	T4_RL1
		Indemnities for Damages	20000
		Quebec Labour Standards Gross	20000
		Quebec Labour Standards Subject	20000
		Quebec Labour Standards Excess	16500
		Quebec Labour Standards Taxable	3500
		Quebec Labour Standards Liability	2.45

Use the BalanceAdjustmentLine.dat file to create the batch line and value records.

```
METADATA|BalanceAdjustmentLine|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|AssignmentNumber|ElementName|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|CA LDG|INDEM_DAM_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/12/31|1|E192278|Quebec Labour Standards Results|CA Tax Reporting Name

METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|InputValueName|EntryValue|ElementName
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/12/31|1|Province|19|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/12/31|1|Quebec Identification Number|7453627101RS0001|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/12/31|1|Reporting Type|T4_RL1|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/12/31|1|Indemnities for Damages|20000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/12/31|1|Quebec Labour Standards Gross|20000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/12/31|1|Quebec Labour Standards Subject|20000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/12/31|1|Quebec Labour Standards Excess|16500|Quebec Labour Standards Results
```

```
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/12/31|1|Quebec Labour Standards Taxable|3500|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation Set|
2020/12/31|1|Quebec Labour Standards Liability|2.45|Quebec Labour Standards Results
```

Example of Loading Quebec Labour Standards Balance Adjustments: Indemnities for Damages – Without Excess Remuneration

This example creates a batch header and group record, with various lines and values to adjust some Quebec Labour Standards balances for one employee - indemnities for damages – without excess remuneration.

At year-end, if any wages are adjusted, you must also adjust the related Quebec Labour Standards balances. In this case, the Indemnities for Damages balance is adjusted by $5,000. Other QLS balances are adjusted as required. In this example, the annual maximum limit is $78,500. No adjustments are required to Gross Earnings.

Balance Name	Indemnities for Damages
YTD Balance Prior to Adjusting	0
Adjustment Value	5,000
YTD Balance After Adjusting	5,000
Notes	Adjusted amount to $5,000. This adds to QLS gross.

Balance Name	Quebec Labour Standards Gross
YTD Balance Prior to Adjusting	70,000
Adjustment Value	5,000
YTD Balance After Adjusting	75,000
Notes	Adjusted since Indemnities for Damages was adjusted, it adds to QLS gross.

Balance Name	Quebec Labour Standards Subject
YTD Balance Prior to Adjusting	70,000
Adjustment Value	5,000
YTD Balance After Adjusting	75,000
Notes	Adjusted to reflect increase in QLS gross.

Balance Name	Quebec Labour Standards Exempt
YTD Balance Prior to Adjusting	0

Adjustment Value	N/A
YTD Balance After Adjusting	0
Notes	No adjustment necessary, no exempt wages.

Balance Name	Quebec Labour Standards Excess
YTD Balance Prior to Adjusting	0
Adjustment Value	N/A
YTD Balance After Adjusting	0
Notes	No adjustment necessary, subject is less than annual limit.

Balance Name	Quebec Labour Standards Taxable
YTD Balance Prior to Adjusting	70,000
Adjustment Value	5,000
YTD Balance After Adjusting	75,000
Notes	Adjusted to reflect subject.

Balance Name	Quebec Labour Standards Liability
YTD Balance Prior to Adjusting	49
Adjustment Value	3.50
YTD Balance After Adjusting	52.50
Notes	Adjusted to reflect calculated value (75,000 * .0007 = 52.50)

Note: When adjusting balances you may also need to adjust any related balances, depending on your business needs. This is simply an example to illustrate the process

Load the Balance Adjustment Header and Group

Use the BalanceAdjustmentHeader.dat file to create the batch header and group records.

```
METADATA|BalanceAdjustmentHeader|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|CA LDG|INDEM_DAM_WO_EXCESS_RENUM

METADATA|BalanceAdjustmentGroup|LegislativeDataGroupName|BatchName|EffectiveDate|PayrollName|
ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|CA LDG|INDEM_DAM_WO_EXCESS_RENUM|2020/12/31|CA Semimonthly|CA Semimonthly
 Consolidation Set|N|N
```

Load the Balance Adjustment Lines and Values

The Balance Adjustment Lines files lines specify the element you will supply values for:

Batch Line Sequence	Assignment Number	Element Name	Tax Reporting Unit Name
1	E192278	Quebec Labour Standards Results	CA Tax Reporting Unit

The Balance Adjustment Value file lines specify the input values for the element named in the line record. The Entry Value is the value for the Input Value Name of the Element Name. For example in the fourth line below, the Entry Value of 5,000 is the value specified for the Indemnities for Damages input value for the Quebec Labour Standards Results element.

You must also specify the contexts for the Province, Quebec Identification Number, and Reporting Type.

Batch Line Sequence	Element Name		
1	Quebec Labour Standards Results	Province	19
		Quebec Identification Number	7453627101RS0001
		Reporting Type	T4_RL1
		Indemnities for Damages	5000
		Quebec Labour Standards Gross	5000
		Quebec Labour Standards Subject	5000
		Quebec Labour Standards Taxable	5000
		Quebec Labour Standards Liability	3.50

Use the BalanceAdjustmentLine.dat file to create the batch line and value records.

```
METADATA|BalanceAdjustmentLine|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|AssignmentNumber|ElementName|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|CA LDG|INDEM_DAM_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|E192278|Quebec Labour Standards Results|ZHRX_CAVS_ST_MULTITRU

METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|InputValueName|EntryValue|ElementName
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Province|19|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Identification Number|7453627101RS0001|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Reporting Type|T4_RL1|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Indemnities for Damages|5000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Gross|5000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Subject|5000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Taxable|5000|Quebec Labour Standards Results
```

```
MERGE|BalanceAdjustmentValue|CA LDG|INDEM_DAM_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Liability|3.50|Quebec Labour Standards Results
```

Example of Loading Quebec Labour Standards Balance Adjustments: Earnings, Exempt Wages – With Excess Remuneration

This example creates a batch header and group record, with various lines and values to adjust some Quebec Labour Standards balances for one employee – exempt wages – with excess remuneration.

At year-end, if any wages are adjusted, you must also adjust the related Quebec Labour Standards balances.

In this case, the several balances are adjusted that reduce subject, and affect QLS gross. Other QLS balances are adjusted as required. In this example, the annual maximum limit is $78,500. No adjustments are required to Gross Earnings.

Balance Name	CCQ Employee Wages
YTD Balance Prior to Adjusting	0
Adjustment Value	5,000
YTD Balance After Adjusting	5,000
Notes	Adjusted amount to $5,000. This is an exempt balance and reduces subject.

Balance Name	RL-1 Box A Adjustment
YTD Balance Prior to Adjusting	0
Adjustment Value	-5,000
YTD Balance After Adjusting	-5,000
Notes	Adjusted amount to $-5,000. This reduces QLS gross.

Balance Name	RL-1 Box R Employment Income Adjustment
YTD Balance Prior to Adjusting	0
Adjustment Value	20,000
YTD Balance After Adjusting	20,000
Notes	Adjusted amount to $20,000. This adds to QLS gross.

Balance Name	Quebec Labour Standards Gross
YTD Balance Prior to Adjusting	70,000

Adjustment Value	15,000
YTD Balance After Adjusting	85,000
Notes	Adjusted for net of $15,000 (of RL-1 Box Adjustments).

Balance Name	Quebec Labour Standards Subject
YTD Balance Prior to Adjusting	70,000
Adjustment Value	10,000
YTD Balance After Adjusting	80,00
Notes	Adjusted to reflect QLS gross – exempt.

Balance Name	Quebec Labour Standards Exempt
YTD Balance Prior to Adjusting	0
Adjustment Value	5,000
YTD Balance After Adjusting	5,000
Notes	Adjusted to reflect exempt wages.

Balance Name	Quebec Labour Standards Excess
YTD Balance Prior to Adjusting	0
Adjustment Value	1,500
YTD Balance After Adjusting	1,500
Notes	Adjusted to reflect amount over the annual limit (80,000 – 78,500).

Balance Name	Quebec Labour Standards Taxable
YTD Balance Prior to Adjusting	70,000
Adjustment Value	8,500
YTD Balance After Adjusting	78,500
Notes	Adjusted to reflect subject, up to limit.

Balance Name	Quebec Labour Standards Liability
YTD Balance Prior to Adjusting	49
Adjustment Value	5.95
YTD Balance After Adjusting	54.95

Notes	Adjusted to reflect calculated value (78,500 * .0007 = 54.95).

Note: When adjusting balances you may also need to adjust any related balances, depending on your business needs. This is simply an example to illustrate the process.

Load the Balance Adjustment Header and Group

Use the BalanceAdjustmentHeader.dat file to create the batch header and group records.

```
METADATA|BalanceAdjustmentHeader|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN

METADATA|BalanceAdjustmentGroup|LegislativeDataGroupName|BatchName|EffectiveDate|PayrollName|
ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|2020/12/31|CA Semimonthly|CA Semimonthly
 Consolidation Set|N|N
```

Load the Balance Adjustment Lines and Values

The Balance Adjustment Lines files lines specify the element you will supply values for:

Batch Line Sequence	Assignment Number	Element Name	Tax Reporting Unit Name
1	E192278	Quebec Labour Standards Results	CA Tax Reporting Unit

The Balance Adjustment Value file lines specify the input values for the element named in the line record. The Entry Value is the value for the Input Value Name of the Element Name. For example in the fourth line below, the Entry Value of 5,000 is the value specified for the CCQ Employee Wages input value for the Quebec Labour Standards Results element.

You must also specify the contexts for the Province, Quebec Identification Number, and Reporting Type.

Batch Line Sequence	Element Name		
1	Quebec Labour Standards Results	Province	19
		Quebec Identification Number	7453627101RS0001
		Reporting Type	T4_RL1
		CCQ Employee Wages	5,000
		RL-1 Box A Adjustment	-5,000
		RL-1 Box R Employment Income Adjustment	20,000
		Quebec Labour Standards Gross	15,000
		Quebec Labour Standards Subject	10,000
		Quebec Labour Standards Exempt	5,000
		Quebec Labour Standards Excess	1,500

Batch Line Sequence	Element Name		
		Quebec Labour Standards Taxable	8,500
		Quebec Labour Standards Liability	5.95

Use the BalanceAdjustmentLine.dat file to create the batch line and value records.

```
METADATA|BalanceAdjustmentLine|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|AssignmentNumber|ElementName|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|E192278|Quebec Labour Standards Results|CA Tax Reporting Unit

METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|InputValueName|EntryValue|ElementName
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Province|19|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Identification Number|7453627101RS0001|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Reporting Type|T4_RL1|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|CCQ Employee Wages|5000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|RL-1 Box A Adjustment|-5000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|RL-1 Box R Employment Income Adjustment|20000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Gross|15000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Subject|10000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Exempt|5000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Excess|1500|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Taxable|8500|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EARN_EXMPT_WAGE_EXCESS_REMUN|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Liability|5.95|Quebec Labour Standards Results
```

Example of Loading Quebec Labour Standards Balance Adjustments: Exempt Wages and Earnings – Without Excess Remuneration

This example creates a batch header and group record, with various lines and values to adjust some Quebec Labour Standards balances for one employee – exempt wages and earnings – without excess remuneration.

In this case, the several exempt balances are adjusted. Other QLS balances are adjusted as required. In this example, the annual maximum limit is $78,500. No adjustments are required to Gross Earnings.

Balance Name	CCQ Employee Wages
YTD Balance Prior to Adjusting	0
Adjustment Value	5,000

YTD Balance After Adjusting	5,000
Notes	Adjusted amount to $5,000. This is an exempt balance and reduces subject.

Balance Name	Heavy Equipment Operator Wages
YTD Balance Prior to Adjusting	0
Adjustment Value	5,000
YTD Balance After Adjusting	5,000
Notes	No changes required because Quebec Labour Standards Exempt was adjusted. **Note:** Only 50% of this amount is considered subject.

Balance Name	Indemnities for Damages
YTD Balance Prior to Adjusting	0
Adjustment Value	5,000
YTD Balance After Adjusting	5,000
Notes	Adjusted amount to $5,000. This adds to QLS gross.

Balance Name	Quebec Labour Standards Gross
YTD Balance Prior to Adjusting	70,000
Adjustment Value	5,000
YTD Balance After Adjusting	75,000
Notes	Adjusted since Indemnities for Damages was adjusted, it adds to QLS gross.

Balance Name	Quebec Labour Standards Subject
YTD Balance Prior to Adjusting	70,000
Adjustment Value	-2,500
YTD Balance After Adjusting	67,500
Notes	Adjustment = Indemnities for Damages (5,000) - CCQ Employee Wages (5,000) - (0.5 * Heavy Equipment Operator Wages (5,000)) = -2,500

Balance Name	Quebec Labour Standards Exempt
YTD Balance Prior to Adjusting	0
Adjustment Value	7,500

YTD Balance After Adjusting	7,500
Notes	Adjusted to reflect exempt wages.

Balance Name	Quebec Labour Standards Excess
YTD Balance Prior to Adjusting	0
Adjustment Value	N/A
YTD Balance After Adjusting	0
Notes	No adjustment necessary, subject is less than annual limit.

Balance Name	Quebec Labour Standards Taxable
YTD Balance Prior to Adjusting	70,000
Adjustment Value	-2,500
YTD Balance After Adjusting	67,500
Notes	Adjusted to reflect subject.

Balance Name	Quebec Labour Standards Liability
YTD Balance Prior to Adjusting	49
Adjustment Value	-1.75
YTD Balance After Adjusting	47.25
Notes	Adjusted to reflect calculated value (67,500 * .0007 = 47.25).

Note: When adjusting balances you may also need to adjust any related balances, depending on your business needs. This is simply an example to illustrate the process.

Load the Balance Adjustment Header and Group

Use the BalanceAdjustmentHeader.dat file to create the batch header and group records.

```
METADATA|BalanceAdjustmentHeader|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM

METADATA|BalanceAdjustmentGroup|LegislativeDataGroupName|BatchName|EffectiveDate|PayrollName|
ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|ZHRX_CAVS_ST_LDGMAIN|EXMP_WG_EARN_WO_EXCESS_RENUM|2020/12/31|CA Semimonthly|CA
 Semimonthly Consolidation Set|N|N
```

Load the Balance Adjustment Lines and Values

The Balance Adjustment Lines files lines specify the element you will supply values for:

Batch Line Sequence	Assignment Number	Element Name	Tax Reporting Unit Name
1	E192278	Quebec Labour Standards Results	CA Tax Reporting Unit

The Balance Adjustment Value file lines specify the input values for the element named in the line record. The Entry Value is the value for the Input Value Name of the Element Name. For example in the fourth line below, the Entry Value of 5,000 is the value specified for the CCQ Employee Wages input value for the Quebec Labour Standards Results element.

You must also specify the contexts for the Province, Quebec Identification Number, and Reporting Type.

Batch Line Sequence	Element Name		
1	Quebec Labour Standards Results	Province	19
		Quebec Identification Number	7453627101RS0001
		Reporting Type	T4_RL1
		CCQ Employee Wages	5000
		Heavy Equipment Operator Wages	5000
		Indemnities for Damages	5000
		Quebec Labour Standards Gross	5000
		Quebec Labour Standards Subject	-2500
		Quebec Labour Standards Exempt	7500
		Quebec Labour Standards Taxable	-2500
		Quebec Labour Standards Liability	-1.75

Use the BalanceAdjustmentLine.dat file to create the batch line and value records.

```
METADATA|BalanceAdjustmentLine|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|AssignmentNumber|ElementName|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|E192278|Quebec Labour Standards Results|CA Tax Reporting Unit

METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|InputValueName|EntryValue|ElementName
MERGE|BalanceAdjustmentValue|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Province|19|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Identification Number|7453627101RS0001|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Reporting Type|T4_RL1|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|CCQ Employee Wages|5000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Heavy Equipment Operator Wages|5000|Quebec Labour Standards Results
```

```
MERGE|BalanceAdjustmentValue|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Indemnities for Damages|5000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Gross|5000|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Subject|-2500|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Exempt|7500|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Taxable|-2500|Quebec Labour Standards Results
MERGE|BalanceAdjustmentValue|CA LDG|EXMP_WG_EARN_WO_EXCESS_RENUM|CA Semimonthly|CA Semimonthly Consolidation
 Set|2020/12/31|1|Quebec Labour Standards Liability|-1.75|Quebec Labour Standards Results
```

Initializing Balances

Overview of Balance Initialization for Canada

In addition to the general guidance provided in the Overview of Balance Initialization topic, consider the following when initializing balances for Canada.

- When initializing Workers' Compensation balances, you must create the account information before processing the balance initialization.
- When initializing Provincial Medical balances, you must create the account information before processing the balance initialization.
- In many cases, specific dimensions are only valid for specific levels of balances. This is primarily the case for provincial level taxes. For example, for balance Provincial Tax Gross, the dimension 'Relationship Province Year to Date' refers to the fact that Province (Geography Area Code of 1) must accompany the balance load to indicate what jurisdiction the Provincial Tax Gross balance is related to. For example, the numerical value of '19' (Area1) refers to Quebec. The following is what the Area context represents for Canada:
    - Area1 identifies the province, entered as the numerical geography code for the province. Refer to the Geography Codes for Canadian Provinces topic.
- If there are associated Hours balances to the base balance (for example, Vacation and Vacation Hours), you must also initialize them. When initializing the Hours balance, the dimension must be the same as the associated base balance.
- When initializing Inception-to-Date (ITD) balances, the Payroll referenced in the batch must exist in the previous year and have payroll periods defined for that previous year. This is because the process attempts to create a balance initialization in the previous year. The process goes as far back as required to distinguish between the dimensions in the batch. For example, if the batch only contains a PTD and an ITD, it needs to go only as far back as the beginning of an earlier period to be able to satisfy the two rows written. If however there are YTD and ITD lines, it uses the first day in the year for YTD, and also needs a day earlier than that to set the ITD.
- Periods other than year-to-date may be required to be loaded, such as period-to-date or inception-to-date. Determination of what additional dimensions are required depends on when the upload activity takes place and the needs of the customer.
- Dimensions other than 'Relationship' may be required to be loaded, such as 'Assignment'. Determination of what additional dimensions are required depends on the level of the element. Taxes are calculated at the relationship level, so they require initialization of dimensions at the 'Relationship' level. Other elements may be created at the assignment levels; therefore dimensions must be initialized at that same level. For example, if an

element is created at the Assignment level, the balance dimension to initialize is at that same level ('Assignment Tax Unit Year to Date' as an example).

- In many cases, specific dimensions are only valid for specific levels of balances. This is primarily the case for provincial level taxes. For example, for balance Provincial Tax Gross, the dimension Relationship Province Year to Date refers to the fact that Province (Geography Area Code of '1') must accompany the balance load to indicate what jurisdiction the Provincial Tax Gross balance is related to. For example, the numerical value of '19' (Area1) refers to Quebec. The following is what the Area context represents for Canada:
    - Area1 identifies the province, entered as the numerical geography code for the province. Refer to the Geography Codes for Canadian Provinces topic.

Legislative or User-defined Context Columns

There are six legislative or user-defined contexts, which must be populated if the balance dimension expects a value for these contexts. For each context, there is a context name, and associated value. For example:

- ContextOneName
- ContextOneValue

Populate the context name attributes with the name of the context usage for that legislative or user-defined context, and the context value attributes with the actual value of that context.

For example, in a case where the Maintenance and Support involuntary deduction balance for the Relationship Tax Unit, Third Party Payee, Reference Code Year to Date dimension requires an additional context, in order to define the reference code, you would populate these values:

Attribute	Value
ContextOneName	Reference Code
ContextOneValue	123456

If the context you are populating is a lookup, populate the context fields in the following manner:

- Context Name: Populate with the lookup type meaning
- Context Value: Populate with the lookup code (not the lookup type meaning)

For example, in one case where you need to populate the ContextOneName and ContextOneValue fields for the dimension Relationship Tax Unit, Statutory Report Type Year to Date, the following attributes should be populated:

Attribute	Value
ContextOneName	Year End Forms
ContextOneValue	T4_RL1

Provincial Medical Example

As another example, in the case where you need to populate the ContextOneName and ContextOneValue fields for the dimension Relationship Tax Unit Province, Reference Code Year to Date for Provincial Medical balances, the following values should be populated:

Attribute	Value
ContextOneName	Reference Code
ContextOneValue	PM_<Provincial Medical Account>

Note: For Quebec ONLY, there is no account for Quebec, so ContextOneValue = PM_UNDEFINED

Workers' Compensation Example

In the case where you need to populate the ContextOneName and ContextOneValue fields for the dimension Relationship Province, Reference Code Year to Date for Workers' Compensation balances, the following values should be populated:

Attribute	Value
ContextOneName	Reference Code
ContextOneValue	WC_<Account number>_<Classification unit>

Related Topics

- Geography Codes for Canadian Provinces
- Balances to Initialize for Canada
- User-Defined Balances for Canada
- Summary Level Balances for Canada
- Miscellaneous Balances for Canada

Balances to Initialize for Canada

The Canada Legislative Balances are balances that Oracle Payroll Cloud uses to perform accurate tax reporting and payroll tax calculations.

Balance initialization is required for the following categories of balances for Canada:

- Legislative balances
- User-defined balances
- Summary balances
- Miscellaneous balances

Canadian Legislative Balances

Canada requires several balances, called Canadian Legislative Balances, to perform accurate tax reporting and payroll tax calculations. For example, balances are used for year-end reporting, Workers' Compensation reporting, and so on. These balances are required to the extent that they apply to the employee. For example, if an employee does not participate in the retirement plan, the required balance Pretax Registered Retirement Plan is not needed for that employee.

Canadian Legislative or User-Defined Context Columns

There are six legislative or user-defined contexts, which must be populated if the balance dimension expects a value for these contexts. These contextual values are used because columns are not available to populate legislative or user-defined data. Instead, they require entry in the user-defined context columns as outlined below. For each context, there is a context name, and value. For example:

- ContextOneName
- ContextOneValue

Populate the context name attributes with the name of the context usage for that legislative or user-defined context, and the context value attributes with the actual value of that context.

Depending on the dimension, Canada may require one of these contextual inputs:

Statutory Report Type

Attribute Name	Value
ContextOneName	Year End Forms
ContextOneValue	[lookup code]. For example T4_RL1

Reference Codes

Workers Compensation

Attribute Name	Value
ContextOneName	Reference Code
ContextOneValue	WC_[Account Number]_[Classification Unit]

Provincial Medical

For Ontario, Manitoba, and Newfoundland and Labrador:

Attribute Name	Value
ContextOneName	Reference Code
ContextOneValue	PM_[Provincial Medical Account]

For Quebec only, there is no account for Quebec:

Attribute Name	Value
ContextOneName	Reference Code
ContextOneValue	PM_UNDEFINED

Involuntary Deductions

Attribute Name	Value
ContextOneName	Reference Code
ContextOneValue	The value of the support order number, or any unique number. For example, 123456.

Pension Registration Number

Used for year-end and element balances

Attribute Name	Value
ContextOneName	Statutory Reporting Code
ContextOneValue	The value of the pension registration number.

If the context you are populating is a lookup, populate the context fields in the following manner:

- Context Name: Populate with the lookup type meaning
- Context Value: Populate with the lookup code (not the lookup type meaning)

Note: In order to help clarify descriptions of the balances, these general statements apply.
- Subject = Gross – Exempt
- Reduced Subject = Subject – Pretax Deductions
- Excess = Reduced Subject – Taxable

Tax is calculated on the **Reduced Subject**. Also, the following applies to limit taxes.

- Reduced Subject – Taxable = Excess

Additionally, in the case of limit taxes (for example, CPP, QPP, EI, QPIP), the taxable and withheld balances are capped by the annual limits. For example, in the case of CPP, the following balances stop at the annual limit:

- CPP Employee Withheld
- Sum of the CPP employee taxable balances
  - CPP Employee Regular Taxabl
  - CPP Employee Nonperiodic Taxable

Statutory Report Type

Secondary classifications in Canada are associated with Statutory Report Types for end of year reporting for both earnings and deductions. These reporting types help determine how the earnings and deductions are reported to

the Canadian Federal and Provincial governments. These reporting types are captured on a flex on the secondary classification. The values for this flex are:

- T4/RL1
- T4A/RL1
- T4A/RL2

The Statutory Report Types are seeded for any secondary classifications that are seeded by the application. For those secondary classifications created by customers, the association to the Reporting Type must be created manually. This association is mandatory.

For the purposes of balance initialization, some of the dimensions require the statutory report type as one of the components of the dimension. This column is populated by use of the Context columns. The required entry in the user-defined context column is as defined as follows, where the Lookup Codes are defined as T4_RL1, T4A_RL1, or T4A_RL2.

Attribute Name	Value
ContextOneName	Year End Forms
ContextOneValue	[Lookup code]

Record of Employment Balances

Initialization of Record of Employment (ROE) balances is not supported. ROEs should be generated from the legacy system after the final payroll is processed, using the 'Change of Service Provider" reason code. If an employee subsequently experiences an interruption of earnings, the ROE will report earnings starting with their first payroll run on Oracle HCM Cloud Payroll.

Vacation Liability Balances

If you process vacation liability, you may initialize the following balances for employees.

Balance Name	Dimension Name
Unprocessed Vacationable Earnings	Relationship Tax Unit Year to Date
Vacation Liability	Relationship Tax Unit, Plan Type Inception to Date
Vacationable Earnings	

For the purposes of balance initialization, some dimensions include a **Plan Type** component. This represents the Vacation Plan and Term. Populate the Plan Type component with the following value:

- VA_<absence plan name>_YYYY_<absence plan ID>

YYYY refers to the vacation term year for which the vacation liability is processed. It is derived on the process/pay date. Employers may accrue liability in one year and allow employees to take vacation in the following year. Alternatively, they may accrue and allow vacation to be taken in the current year.

For example, VA_Vacation Plan_2018_123456789098765.

Tax Balances for Prior Contributions

Oracle Cloud contains tax balances to hold the prior contribution values for an employee. This is to accommodate those values deducted by a previous employer. These prior values are considered for the annual maximum contribution limits. From the employee's perspective, these balances will not appear in the current year's YTD amounts, nor will they appear on any year end forms.

The tax balances are for Prior Canada Pension Plan (CPP), Prior Quebec Pension Plan (QPP), Prior Quebec Parental Insurance Plan (QPIP), and Prior Employment Insurance (EI). The balances and the required dimensions are listed in this table and should only be initialized if you need to account for prior balances. Examples include prior balances deducted in the case of acquisitions, amalgamations, or a change in business status.

Employment Insurance (EI)

Balance Name	Dimension Name
Prior EI Employee Taxable	Relationship No Calculation Breakdown, Province, Statutory Report Type Year to Date
Prior EI Employee Withheld	
Prior EI Employer Taxable	
Prior EI Employer Liability	

Canada Pension Plan (CPP)

Balance Name	Dimension Name
Prior CPP Employee Basic Exemption	Relationship No Calculation Breakdown, Statutory Report Type Year to Date
Prior CPP Employee Taxable	
Prior CPP Employee Withheld	
Prior CPP Employer Basic Exemption	
Prior CPP Employer Taxable	
Prior CPP Employer Liability	

Quebec Pension Plan (QPP)

Balance Name	Dimension Name
Prior QPP Employee Basic Exemption	Relationship No Calculation Breakdown, Statutory Report Type Year to Date
Prior QPP Employee Taxable	
Prior QPP Employee Withheld	
Prior QPP Employer Basic Exemption	
Prior QPP Employer Taxable	

Balance Name	Dimension Name
Prior QPP Employer Liability	

Quebec Parental Insurance Plan (QPIP)

Balance Name	Dimension Name
Prior QPIP Employee Taxable	Relationship No Calculation Breakdown, Statutory Report Type Year to Date
Prior QPIP Employee Withheld	
Prior QPIP Employer Taxable	
Prior QPIP Employer Liability	

Quebec Labour Standards Balances

Employers with employees working in Quebec are subject to the Quebec Labour Standards (QLS) levy. Initialize these balances only if applicable.

If you are going live at any point other than the start of the tax year, for accurate QLS calculations and reporting, you must initialize the balances prior to running payrolls. Alternatively, you may perform balance adjustments at the end of the year.

For the purposes of balance initialization, the dimension required for QLS balances includes a **Statutory Report Code** component. This represents the **Quebec Identification Number** associated with the balance, and it is mandatory to populate.

Note: Depending on your unique requirements, you may require other balances to be initialized. You may need to initialize other balances separately that are part of QLS gross or QLS exempt earnings. Please see the Use Cases and Example Files section for QLS examples.

For additional information on the Quebec Labour Standards Report, refer to these documents:

Cloud Readiness What's New Document:

https://www.oracle.com/webfolder/technetwork/tutorials/tutorial/cloud/r13/wn/wfr/releases/21A/21A-wf-rewards-wn.htm#F16474

Administering Payroll for Canada Guide:

https://docs.oracle.com/en/cloud/saas/human-resources/22c/faapc/index.html

Balance Name	Dimension Name	Notes
Quebec Labour Standards Gross	Relationship No Calculation Breakdown, Statutory Report Type Year to Date	The sum of these balances: Gross Earnings RL-1 Box A Adjustment RL-1 Box R Employment Income Adjustment Indemnities in Lieu of Notice Indemnities for Damages

Balance Name	Dimension Name	Notes
		RL-1 Deferred Salary or Wages If any of these balances are initialized or adjusted, you must also adjust the Quebec Labour Standards Gross balance accordingly.
Quebec Labour Standards Subject	Relationship Tax Unit, Province, Statutory Report Type, Statutory Report Code Year to Date	The Quebec Labour Standards Gross - sum of these balances (which are exempt balances): RL-1 Employee Benefit Plan RL-1 Employee Trust Directors Fees Benefit for Amount Paid for Shares Commission or Committee Member Fees Quebec Employees Outside Canada Wages Parity Committee Employee Wages CCQ Employee Wages Heavy Equipment Operator Wage * .05 (Percent Remuneration not Subject) Other Labour Standards Exempt Remuneration
Quebec Labour Standards Exempt		Quebec Labour Standards Gross - Quebec Labour Standards Subject. Note: Although this balance is maintained, it is not displayed on the Quebec Labour Standards Report.
Quebec Labour Standards Taxable		Quebec Labour Standards Subject, not to exceed the annual maximum limit. For example, the 2020 limit is $78,500.
Quebec Labour Standards Excess		Quebec Labour Standards Subject - Annual Maximum Limit. This represents any amount of the Quebec Labour Standards Subject in excess of the annual maximum limit.
Quebec Labour Standards Liability		The amount calculated by applying the annual rate to the Quebec Labour Standards Taxable.
Benefit from Amount Paid for Shares		N/A
CCQ Employee Wages		N/A
Commission or Committee Member Fees		N/A
Director's Fees		N/A
Heavy Equipment Operator Wages		N/A

Balance Name	Dimension Name	Notes
Indemnities for Damages		N/A
Indemnities in Lieu of Notice		N/A
Parity Committee Employee Wages		N/A
Other Labour Standards Exempt Remuneration		N/A

Tax and Year-End Balances

Please refer to *Geography Codes for Canadian Provinces* for the tax and year-end balances available to load for Canada.

Related Topics

- Overview of Balance Initialization
- Steps to Initialize Balances

User-Defined Balances for Canada

User-defined balances are those balances typically associated with an element defined at the time of implementation and are customer specific.

These balances are needed primarily to support the proper reflection and reference ability of balances within the application. You may need to initialize balances for any of these user-defined elements for those balances that you want to create and maintain within the application. A listing of the seeded primary and secondary classifications is noted in the following section to help determine what associated balances may be initialized.

Each user-defined element should be reviewed to see if the associated balances for that user-defined element must be initialized.

Summary Level Balances for Canada

Summary level balances are those balances that are associated with a grouping of other balances, or a total of other balances.

They are associated with a primary or secondary classification that relates to the elements. The primary or secondary classification contains an associated balance name that is named the same as the classification. For example, under the Involuntary Deductions primary classification, the secondary classification of Garnishments exists, as does the balance of the Garnishments and Involuntary Deductions. If balances at these levels are required by the customer, the balances associated with these classifications may be initialized. For example, the 'Pretax Deductions' balance may require initialization, which may be the balance for all the elements whose classification is Pretax Deductions. Similarly, the 'Union Dues' balance may require initialization, which may be the summary level balance for the Pretax/Union Dues primary/secondary classifications.

If you are initializing these element level balances, only that specific balance will be initialized. Summary level balances are not initialized automatically. For example, if you are initializing 'Company X Union Dues', the summary level balances of 'Company X Union Dues' or 'Pretax Deductions' are not automatically initialized when the element level balance is

initialized. In order to initialize these summary level balances, you have the option to initialize summary level balances along with the other balances, or you must manually add additional balance feeds to the summary level balances.

Secondary Classification Balances

This table contains all the secondary classification balances that are associated to the primary classification elements that may be initialized.

> **Note:** Do not create elements or balances with the exact same name as the summary level balances as this causes conflict when the process tries to derive the balance feeds for the initialization elements.

Involuntary Deductions

Secondary classification balances for the Involuntary Deductions primary classification:

Secondary Classification Balance	Dimension
Garnishments	Relationship Tax Unit, Province, Statutory Report Type Year to Date
Maintenance and Support	
Tax Levy	

Nonpayroll Payments

Secondary classification balances for the Nonpayroll Payments primary classification:

Secondary Classification Balance	Dimension
Expense Reimbursement	Assignment Tax Unit Year to Date

Pretax Deductions

Secondary classification balances for the Pretax Deductions primary classification:

Secondary Classification Balance	Dimension
Registered Retirement Plan **Note:** Balance is Pretax Registered Retirement Plan	Relationship Tax Unit, Province, Statutory Report Type Year to Date
Union Dues **Note:** Balance is Pretax Union Dues	

Standard Earnings

Secondary classification balances for the Standard Earnings primary classification:

Secondary Classification Balance	Dimension
Overtime	Relationship Tax Unit, Province, Statutory Report Type Year to Date
Regular	
Shift	
Statutory Holiday	

Supplemental Earnings

Secondary classification balances for the Supplemental Earnings primary classification:

Secondary Classification Balance	Dimension
Awards and Prizes	Relationship Tax Unit, Province, Statutory Report Type Year to Date
Benefit Plan Credits	
Bonus	
Commission	
Jury Duty Pay	
Pay in Lieu of Notice	
Pensions and Annuities	
Retiring Allowance Non-taxable	
Retiring Allowance Taxable	
Sick Pay	
Vacation Pay	

Taxable Benefits

Secondary classification balances for the Taxable Benefits primary classification:

Secondary Classification Balance	Dimension
Board and Lodging	Relationship Tax Unit, Province, Statutory Report Type Year to Date
Gifts and Awards	
Group Term Life Insurance	
Personal Use of Company Car	
Provincial Medical	
Private Health Services Plan	

Secondary Classification Balance	Dimension
Registered Retirement Savings Plan	

Voluntary Deductions

Secondary classification balances for the Voluntary Deductions primary classification:

Secondary Classification Balance	Dimension
Benefits	Relationship Tax Unit, Province, Statutory Report Type Year to Date
Employer Reimbursements	
Loans	
Professional Fees	
Voluntary Registered Retirement Plan	
Union Dues	

Primary Classification Balances

Summary level balances also exist for primary classifications that may require initialization. For example, the 'Pretax Deductions' balance may require initialization, which may be the balance for all the elements whose classification is Pretax Deductions.

This table contains all the primary classification balances that are associated to the primary classification elements that may be initialized.

Primary Classification Balances	Dimension	Definition
Absences	Relationship Tax Unit, Province, Statutory Report Type Year to Date	Total of all elements whose classification is Absences.
Deductions		Total of all elements whose classification is Pretax Deductions, Employee Tax Deductions, Involuntary Deductions, and Voluntary Deductions.
Employer Liabilities		Total of all elements whose classification is Employer Liabilities.
Employee Tax Deductions		Total of all elements whose classification is Employee Tax Deductions.
Employer Taxes		Total of all elements whose classification is Employer Taxes.
Involuntary Deductions		Total of all elements whose classification is Involuntary Deductions.
Non-payroll Payments		Total of all elements whose classification is Nonpayroll Payment.

Primary Classification Balances	Dimension	Definition
Pretax Deductions		Total of all elements whose classification is Pretax Deductions.
Standard Earnings		Total of all elements whose classification is Standard Earnings.
Supplemental Earnings		Total of all elements whose classification is Supplemental Earnings.
Taxable Benefits		Total of all elements whose classification is Taxable Benefits.
Voluntary Deductions		Total of all elements whose classification is Voluntary Deductions.

Miscellaneous Balances for Canada

There are some additional balances to consider that may not be a summary-level balance mentioned in the previous topic.

They may be balances that are associated to elements that are created along with the base element (shadow elements).

These balances listed in this table may be used to initialize limits or other important user-defined balances and are important to take into consideration.

Balance Name	Dimension	Definition
Taxable Benefit Paid in Kind	Relationship Tax Unit, Province, Statutory Report Type Year to Date	The value of Quebec taxable benefits provided to the employee when no remuneration was paid.
Taxable Benefit without Remuneration	Relationship Tax Unit, Province, Statutory Report Type Year to Date	The value of federal taxable benefits provided to an employee when no remuneration was paid.

Related Topics

- Overview of Balance Initialization
- Steps to Initialize Balances

Example of Loading Quebec Employee's YTD Federal and Provincial Balances

In this example, you create balance initialization records to initialize an employee's YTD federal and provincial-level balances for a Quebec employee.

When initializing year-to-date balances, you also initialize any related balances. For example, when initializing Federal Tax Withheld balance, other related balances may need to be initialized as well (for example, Federal Exempt, Federal Tax Not Taken, etc.), depending on your business needs.

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|LegislativeDataGroupName|BatchName|UploadDate
MERGE|InitializeBalanceBatchHeader|CA LDG|CA Bal Init|2019/01/31
```

Load the Initialize Balance Batch Lines

In the following example a number of balances are being initialized for the same employee. The employee is identified by their payroll relationship number, employment terms number and assignment number.

Attribute	Value
PayrollRelationshipNumber	8182639
TermNumber	ET8182639
AssignmentNumber	E182639

The employee's payroll and tax reporting unit must also be specified on every initialize balance line

Attribute	Value
PayrollName	Vision CA Weekly
TaxUnitName	CA Tax Reporting Unit

Load these details to initialize the YTD Federal Tax Withheld to 6338.94:

Attribute	Value
LineSequence	1
BalanceName	Federal Tax Withheld
DimensionName	Relationship Tax Unit, Province, Statutory Report Type Year to Date
AreaOne	19
Value	6338.94
ContextOneName	Year End Forms
ContextOneValue	T4_RL1

Load these details to initialize the YTD Provincial Tax Withheld balance to 7025.80:

Attribute	Value
LineSequence	2
BalanceName	Provincial Tax Withheld
DimensionName	Relationship Tax Unit, Province, Statutory Report Type Year to Date
AreaOne	19
Value	7025.80
ContextOneName	Year End Forms
ContextOneValue	T4_RL1

Load these details to initialize the YTD QPP Employee Withheld balance:

Attribute	Value
LineSequence	3
BalanceName	QPP Employee Withheld
DimensionName	Relationship Tax Unit, Province, Statutory Report Type Year to Date
AreaOne	19
Value	1559.68
ContextOneName	Year End Forms
ContextOneValue	T4_RL1

Load these details to initialize the YTD EI Employee Withheld balance:

Attribute	Value
LineSequence	4
BalanceName	EI Employee Withheld
DimensionName	Relationship Tax Unit, Province, Statutory Report Type Year to Date
AreaOne	19
Value	459.00
ContextOneName	Year End Forms
ContextOneValue	T4_RL1

Load these details to initialize the YTD EI Employer Liability balance:

Attribute	Value
LineSequence	5
BalanceName	EI Employer Liability
DimensionName	Relationship Tax Unit, Province, Statutory Report Type Year to Date
AreaOne	19
Value	646.80
ContextOneName	Year End Forms
ContextOneValue	T4_RL1

Load these details to initialize the YTD Workers Compensation Gross balance:

Attribute	Value
LineSequence	6
BalanceName	Workers Compensation Gross
DimensionName	Relationship Tax Unit, Province, Reference Code Year to Date
AreaOne	19
Value	30000.00
ContextOneName	Reference Code
ContextOneValue	WC_QC1234_QCCU1234

Load these details to initialize the YTD Workers Compensation Liability balance:

Attribute	Value
LineSequence	7
BalanceName	Workers Compensation Liability
DimensionName	Relationship Tax Unit, Province, Reference Code Year to Date
AreaOne	19
Value	435.00
ContextOneName	Reference Code
ContextOneValue	WC_QC1234_QCCU1234

Load these details to initialize the YTD Provincial Medical Gross balance:

Attribute	Value
LineSequence	8
BalanceName	Provincial Medical Gross
DimensionName	Relationship Tax Unit, Province, Reference Code Year to Date
AreaOne	19
Value	30000.00
ContextOneName	Reference Code
ContextOneValue	PM_UNDEFINED

Load these details to initialize the YTD Provincial Medical Liability balance:

Attribute	Value
LineSequence	9
BalanceName	Provincial Medical Liability
DimensionName	Relationship Tax Unit, Province, Reference Code Year to Date
AreaOne	19
Value	1278.00
ContextOneName	Reference Code
ContextOneValue	PM_UNDEFINED

Use the InitializeBalanceBatchLine.dat file to load the balance initialization lines defined above:

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
AreaOne|ContextOneName|ContextOneValue|Value

MERGE|InitializeBalanceBatchLine|CA LDG|CA Bal Init|1|8182639|ET8182639|E8182639|Vision CA Weekly|CA Tax
 Reporting Unit|Federal Tax Withheld|Relationship Tax Unit, Province, Statutory Report Type Year to Date|19|
Year End Forms|T4_RL1|6338.94
MERGE|InitializeBalanceBatchLine|CA LDG|CA Bal Init|2|8182639|ET8182639|E8182639|Vision CA Weekly|CA Tax
 Reporting Unit|Provincial Tax Withheld|Relationship Tax Unit, Province, Statutory Report Type Year to Date|
19|Year End Forms|T4_RL1|7025.80
MERGE|InitializeBalanceBatchLine|CA LDG|CA Bal Init|3|8182639|ET8182639|E8182639|Vision CA Weekly|CA Tax
 Reporting Unit|QPP Employee Withheld|Relationship Tax Unit, Province, Statutory Report Type Year to Date|
19|Year End Forms|T4_RL1|1559.68
MERGE|InitializeBalanceBatchLine|CA LDG|CA Bal Init|4|8182639|ET8182639|E8182639|Vision CA Weekly|CA Tax
 Reporting Unit|EI Employee Withheld|Relationship Tax Unit, Province, Statutory Report Type Year to Date|19|
Year End Forms|T4_RL1|459.00
MERGE|InitializeBalanceBatchLine|CA LDG|CA Bal Init|5|8182639|ET8182639|E8182639|Vision CA Weekly|CA Tax
 Reporting Unit|EI Employer Liability|Relationship Tax Unit, Province, Statutory Report Type Year to Date|
19|Year End Forms|T4_RL1|646.80
MERGE|InitializeBalanceBatchLine|CA LDG|CA Bal Init|6|8182639|ET8182639|E8182639|Vision CA Weekly|CA Tax
 Reporting Unit|Workers Compensation Gross|Relationship Tax Unit, Province, Reference Code Year to Date|19|
Reference Code|WC_QC1234_QCCU1234|30000.00
```

```
MERGE|InitializeBalanceBatchLine|CA LDG|CA Bal Init|7|8182639|ET8182639|E8182639|Vision CA Weekly|CA Tax
 Reporting Unit|Workers Compensation Liability|Relationship Tax Unit, Province, Reference Code Year to Date|
19|Reference Code|WC_QC1234_QCCU1234|435.00
MERGE|InitializeBalanceBatchLine|CA LDG|CA Bal Init|8|8182639|ET8182639|E8182639|Vision CA Weekly|CA Tax
 Reporting Unit|Provincial Medical Gross|Relationship Tax Unit, Province, Reference Code Year to Date|19|
Reference Code|PM_UNDEFINED|30000.00
MERGE|InitializeBalanceBatchLine|CA LDG|CA Bal Init|9|8182639|ET8182639|E8182639|Vision CA Weekly|CA Tax
 Reporting Unit|Provincial Medical Liability|Relationship Tax Unit, Province, Reference Code Year to Date|
19|Reference Code|PM_UNDEFINED|1278.00
```

Note: When initializing specific balances, you may also need to initialize any related balances. This is simply an example to illustrate the process. For example, when initializing Federal Tax Withheld balance, related balances such as Federal Tax Subject and Federal Tax Gross may also need to be initialized, depending on your business needs.

Related Topics

- Overview of Balance Initialization
- Steps to Initialize Balances
- Overview of Balance Initialization for Canada

Example of Loading Quebec Labour Standards Balances: Remuneration Subject to Contribution below the Maximum Limit – With Exempt Wages

In this example, you create balance initialization records to initialize an employee's QLS balances where the wages are below the 2020 annual maximum limit of $78,500 and exempt wages do exist.

Note: These are only examples and you may require other balances to be initialized depending on your unique requirements. You may need to initialize other balances separately that are part of QLS gross or QLS exempt earnings.

Balance Name	Value	Notes
Gross Earnings	6,000	N/A
Quebec Labour Standards Gross	6,000	N/A
Other Labour Standards Exempt Remuneration	3,000	N/A
Quebec Labour Standards Subject	3,000	There are Quebec Labour Standard Exempt Wages, so Subject = Gross – Exempt Wages
Quebec Labour Standards Exempt	3,000	Gross – Subject
Quebec Labour Standards Taxable	3,000	Subject amount of 3,000 is Taxable since it is under the annual limit
Quebec Labour Standards Liability	2.10	Calculated value of Taxable * rate. For example, using the values for 2020: 3,000 * 0.0007 = 2.10

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|QLS_REM_BELOW_MAX_WTH|2020/01/01|CA LDG
```

Load the Initialize Balance Batch Lines

Use the InitializeBalanceBatchLine.dat file to create the batch lines.

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
Value|AreaOne|ContextOneName|ContextOneValue|ContextTwoName|ContextTwoValue
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WTH|1|192259|ET192259|E192259|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Gross|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date|6000.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WTH|2|192259|ET192259|E192259|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Subject|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date|3000.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WTH|3|192259|ET192259|E192259|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Taxable|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date|3000.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WTH|4|192259|ET192259|E192259|CA Semimonthly|CA
 Tax Reporting Unit|Quebec Labour Standards Liability|RelationshipTax Unit, Province, Statutory Report Type,
 Statutory Report Code Year to Date|2.10|19|Year End Forms|T4_RL1|Statutory Reporting Code|7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WTH|5|192259|ET192259|E192259|CA Semimonthly|CA
 Tax Reporting Unit|Gross Earnings|Relationship Tax Unit, Province, Statutory Report Type Year to Date|
6000.00|19|Year End Forms|T4_RL1||
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WTH|6|192259|ET192259|E192259|CA Semimonthly|CA
 Tax Reporting Unit|Other Labour Standards Exempt Remuneration|Relationship Tax Unit, Province, Statutory
 Report Type Year to Date|3000.00|19|Year End Forms|T4_RL1||
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WTH|7|192259|ET192259|E192259|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Exempt|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date|3000.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
```

Example of Loading Quebec Labour Standards Balances: Remuneration Subject to Contribution below the Maximum Limit Without Exempt Wages

In this example, you create balance initialization records to initialize an employee's QLS balances where the wages are below the 2020 annual maximum limit of $78,500 and no exempt wages exist.

> **Note:** These are only examples and you may require other balances to be initialized depending on your unique requirements. You may need to initialize other balances separately that are part of QLS gross or QLS exempt earnings.

Balance Name	Value	Notes
Gross Earnings	6,000	N/A

Balance Name	Value	Notes
Quebec Labour Standards Gross	6,000	N/A
Quebec Labour Standards Subject	6,000	There are no Quebec Labour Standard Exempt Wages, so Subject = Gross.
Quebec Labour Standards Taxable	6,000	Subject amount of 6,000 is Taxable since it is under the annual limit.
Quebec Labour Standards Liability	4.20	Calculated value of Taxable * rate. For example, using the values for 2020: 6,000 * 0.0007 = 4.20.

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|QLS_REM_BELOW_MAX_WO|2020/01/01|CA LDG
```

Load the Initialize Balance Batch Lines

Use the InitializeBalanceBatchLine.dat file to create the batch lines.

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|BalanceName|DimensionName|Value|PayrollName|AreaOne|
TaxUnitName|ContextOneName|ContextOneValue|ContextTwoName|ContextTwoValue
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WO|1|192191|ET192191|E192191|Quebec Labour
 Standards Gross|Relationship Tax Unit, Province, Statutory Report Type, Statutory Report Code Year to
 Date|6000.00|CA Semimonthly|19|CA Tax Reporting Unit|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WO|2|192191|ET192191|E192191|Quebec Labour
 Standards Subject|Relationship Tax Unit, Province, Statutory Report Type, Statutory Report Code Year
 to Date|6000.00|CA Semimonthly|19|CA Tax Reporting Unit|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WO|3|192191|ET192191|E192191|Quebec Labour
 Standards Taxable|Relationship Tax Unit, Province, Statutory Report Type, Statutory Report Code Year
 to Date|6000.00|CA Semimonthly|19|CA Tax Reporting Unit|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WO|4|192191|ET192191|E192191|Quebec Labour
 Standards Liability|Relationship Tax Unit, Province, Statutory Report Type, Statutory Report Code Year
 to Date|4.20|CA Semimonthly|19|CA Tax Reporting Unit|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_BELOW_MAX_WO|5|192191|ET192191|E192191|Gross Earnings|
Relationship Tax Unit, Province, Statutory Report Type Year to Date|6000.00|CA Semimonthly|19|CA Tax
 Reporting Unit|Year End Forms|T4_RL1||
```

Example of Loading Quebec Labour Standards Balances: Remuneration Subject to Contribution above the Maximum Limit With Exempt Wages

In this example, you create balance initialization records to initialize an employee's QLS balances where the wages are above the 2020 annual maximum limit of $78,500 and exempt wages do exist.

Note: These are only examples and you may require other balances to be initialized depending on your unique requirements. You may need to initialize other balances separately that are part of QLS gross or QLS exempt earnings.

Balance Name	Value	Notes
Gross Earnings	90,000	N/A
Quebec Labour Standards Gross	90,000	N/A
Other Labour Standards Exempt Remuneration	5,000	N/A
Quebec Labour Standards Subject	85,000	There are Quebec Labour Standard Exempt Wages, so Subject = Gross – Exempt Wages.
Quebec Labour Standards Exempt	5,000	Gross – Subject
Quebec Labour Standards Excess	6,500	Subject – Limit. For example, using the values for 2020: 85,000 – 78,500 = 6,500
Quebec Labour Standards Taxable	78,500	Subject amount up to the limit. For example, using the values for 2020: Subject is 85,000 but the limit = 78,500 so Taxable = 78,500
Quebec Labour Standards Liability	54.95	Calculated value of Taxable * rate. For example, using the values for 2020: 78,500 * 0.0007 = 54.95

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|QLS_REM_ABOVE_MAX_WTH|2020/01/01|CA LDG
```

Load the Initialize Balance Batch Lines

Use the InitializeBalanceBatchLine.dat file to create the batch lines.

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
Value|AreaOne|ContextOneName|ContextOneValue|ContextTwoName|ContextTwoValue
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WTH|1|192213|ET192213|E192213|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Gross|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date| 90000 |19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WTH|2|192213|ET192213|E192213|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Subject|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date| 85000 |19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WTH|3|192213|ET192213|E192213|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Taxable|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date| 78500 |19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WTH|4|192213|ET192213|E192213|CA Semimonthly|CA
 Tax Reporting Unit|Quebec Labour Standards Liability|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date| 54.95 |19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WTH|5|192213|ET192213|E192213|CA Semimonthly|CA
 Tax Reporting Unit|Gross Earnings|Relationship Tax Unit, Province, Statutory Report Type Year to Date|
 90000 |19|Year End Forms|T4_RL1||
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WTH|6|192213|ET192213|E192213|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Excess|Relationship Tax Unit, Province, Statutory Report
```

```
 Type, Statutory Report Code Year to Date| 6500 |19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WTH|7|192213|ET192213|E192213|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Exempt|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date| 5000 |19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WTH|8|192213|ET192213|E192213|CA Semimonthly|CA
 Tax Reporting Unit|Other Labour Standards Exempt Remuneration|Relationship Tax Unit, Province, Statutory
 Report Type Year to Date| 5000 |19|Year End Forms|T4_RL1||
```

Example of Loading Quebec Labour Standards Balances: Remuneration Subject to Contribution above the Maximum Limit Without Exempt Wages

In this example, you create balance initialization records to initialize an employee's QLS balances where the wages are above the 2020 annual maximum limit of $78,500 and no exempt wages exist.

Note: These are only examples and you may require other balances to be initialized depending on your unique requirements. You may need to initialize other balances separately that are part of QLS gross or QLS exempt earnings.

Balance Name	Value	Notes
Gross Earnings	90,000	N/A
Quebec Labour Standards Gross	90,000	N/A
Quebec Labour Standards Subject	90,000	There are no Quebec Labour Standard Exempt Wages, so Subject = Gross
Quebec Labour Standards Excess	11,500	Subject – Limit. For example, using the values for 2020: 90,000 – 78,500 = 11,500
Quebec Labour Standards Taxable	78,500	Subject amount up to the limit. For example, using the values for 2020: Subject is 90,000 but the limit = 78,500 so Taxable = 78,500
Quebec Labour Standards Liability	54.95	Calculated value of Taxable * rate. For example, using the values for 2020: 6,000 * 0.0007 = 54.95

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|QLS_REM_ABOVE_MAX_WO|2020/01/01|CA LDG
```

Load the Initialize Balance Batch Lines

Use the InitializeBalanceBatchLine.dat file to create the batch lines.

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
Value|AreaOne|ContextOneName|ContextOneValue|ContextTwoName|ContextTwoValue
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WO|1| 192262 |ET192262|E192262|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Gross|Relationship Tax Unit, Province, Statutory Report
```

```
 Type, Statutory Report Code Year to Date|90000.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WO|2| 192262 |ET192262|E192262|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Subject|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date|90000.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WO|3| 192262 |ET192262|E192262|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Taxable|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date|78500.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WO|4| 192262 |ET192262|E192262|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Liability|Relationship Tax Unit, Province, Statutory
 Report Type, Statutory Report Code Year to Date|54.95|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WO|5| 192262 |ET192262|E192262|CA Semimonthly|
CA Tax Reporting Unit|Gross Earnings|Relationship Tax Unit, Province, Statutory Report Type Year to Date|
90000.00|19|Year End Forms|T4_RL1||
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_REM_ABOVE_MAX_WO|6| 192262 |ET192262|E192262|CA Semimonthly|
CA Tax Reporting Unit|Quebec Labour Standards Excess|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date|11500.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
```

Example of Quebec Labour Standards Balances: Initializing Director's Fees

In this example, you create balance initialization records to initialize an employee's QLS balances, including Director's Fees.

In this example, wages are below the 2020 annual maximum limit of $78,500. This example illustrates the relationship between QLS balances.

> **Note:** These are only examples and you may require other balances to be initialized depending on your unique requirements. You may need to initialize other balances separately that are part of QLS gross or QLS exempt earnings.

Balance Name	Value	Notes
Gross Earnings	6,000	N/A
Quebec Labour Standards Gross	6,000	N/A
Directors Fees	1,000	Directors Fees balance are exempt wages. The balance that represents the "Directors' fees" on form. You may include this balance used to calculate the liability. **Note:** If you initialize this amount, it should also be reduced from the Quebec Labour Standards Subject amount, which in turn reduces the Quebec Labour Standards Taxable amount.
Quebec Labour Standards Subject	5,000	There are Quebec Labour Standard Exempt Wages, so Subject = Gross – Exempt Wages
Quebec Labour Standards Taxable	5,000	Subject amount of 5,000 is Taxable since it is under the annual limit

Balance Name	Value	Notes
Quebec Labour Standards Liability	3.5	Calculated value of Taxable * rate. For example, using the values for 2020: 5,000 * 0.0007 = 3.5

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|QLS_DIR_FEES|2020/01/01|CA LDG
```

Load the Initialize Balance Batch Lines

Use the InitializeBalanceBatchLine.dat file to create the batch lines.

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|Value|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
AreaOne|ContextOneName|ContextOneValue|ContextTwoName|ContextTwoValue
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_DIR_FEES|1|6000.00|192191|ET192191|E192191|CA Semimonthly|CA
 Tax Reporting Unit|Quebec Labour Standards Gross|Relationship Tax Unit, Province, Statutory Report Type,
 Statutory Report Code Year to Date|19|Year End Forms|T4_RL1|Statutory Reporting Code|7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_DIR_FEES|2|5000.00|192191|ET192191|E192191|CA Semimonthly|CA
 Tax Reporting Unit|Quebec Labour Standards Subject|Relationship Tax Unit, Province, Statutory Report Type,
 Statutory Report Code Year to Date|19|Year End Forms|T4_RL1|Statutory Reporting Code|7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_DIR_FEES|3|5000.00|192191|ET192191|E192191|CA Semimonthly|CA
 Tax Reporting Unit|Quebec Labour Standards Taxable|Relationship Tax Unit, Province, Statutory Report Type,
 Statutory Report Code Year to Date|19|Year End Forms|T4_RL1|Statutory Reporting Code|7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_DIR_FEES|4|3.50|192191|ET192191|E192191|CA Semimonthly|CA Tax
 Reporting Unit|Quebec Labour Standards Liability|Relationship Tax Unit, Province, Statutory Report Type,
 Statutory Report Code Year to Date|19|Year End Forms|T4_RL1|Statutory Reporting Code|7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_DIR_FEES|5|6000.00|192191|ET192191|E192191|CA Semimonthly|CA Tax
 Reporting Unit|Gross Earnings|Relationship Tax Unit, Province, Statutory Report Type Year to Date|19|Year
 End Forms|T4_RL1||
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_DIR_FEES|6|1000.00|192191|ET192191|E192191|CA Semimonthly|CA Tax
 Reporting Unit|Directors Fees|Relationship Tax Unit, Province, Statutory Report Type Year to Date|19|Year
 End Forms|T4_RL1|Statutory Reporting Code|7453627101RS0001
```

Example of Quebec Labour Standards: Initializing RL1 Box A Adjustment Balance – Above the Maximum Limit

In this example, you create balance initialization records to initialize an employee's QLS balances, including the RL-1 Box A Adjustment balance, above the maximum limit.

In this example, wages are above the 2020 annual maximum limit of $78,500 and no exempt wages exist. This example illustrates the relationship between QLS balances.

Note: These are only examples and you may require other balances to be initialized depending on your unique requirements. You may need to initialize other balances separately that are part of QLS gross or QLS exempt earnings.

Balance Name	Value	Notes
Gross Earnings	70,000	N/A

Balance Name	Value	Notes
RL-1 Box A Adjustment	20,000	N/A
Quebec Labour Standards Gross	90,000	N/A
Quebec Labour Standards Subject	90,000	There are no Quebec Labour Standard Exempt Wages, so Subject = Gross
Quebec Labour Standards Excess	11,500	Subject – Limit. For example, using the values for 2020: 90,000 – 78,500 = 11,500
Quebec Labour Standards Taxable	78,500	Subject amount up to the limit. For example, using the values for 2020: Subject is 90,000 but the limit = 78,500 so Taxable = 78,500
Quebec Labour Standards Liability	54.95	Calculated value of Taxable * rate. For example, using the values for 2020: 78,500 * 0.0007 = 54.95

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|QLS_RL1_ABOVE|2020/01/01|CA LDG
```

Load the Initialize Balance Batch Lines

Use the InitializeBalanceBatchLine.dat file to create the batch lines.

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
Value|AreaOne|ContextOneName|ContextOneValue|ContextTwoName|ContextTwoValue
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_RL1_ABOVE|1|192191|ET192191|E192191|CA Semimonthly|CA Tax
 Reporting Unit|Gross Earnings|Relationship Tax Unit, Province, Statutory Report Type Year to Date|70000.00|
19|Year End Forms|T4_RL1||
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_RL1_ABOVE|2|192191|ET192191|E192191|CA Semimonthly|CA Tax
 Reporting Unit|RL-1 Box A Adjustment|Relationship Tax Unit Year to Date|20000.00|19|Year End Forms|T4_RL1|
Statutory Reporting Code|7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_RL1_ABOVE|3|192191|ET192191|E192191|CA Semimonthly|CA
 Tax Reporting Unit|Quebec Labour Standards Gross|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date|90000.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_RL1_ABOVE|4|192191|ET192191|E192191|CA Semimonthly|CA
 Tax Reporting Unit|Quebec Labour Standards Subject|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date|90000.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_RL1_ABOVE|5|192191|ET192191|E192191|CA Semimonthly|CA
 Tax Reporting Unit|Quebec Labour Standards Excess|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date|11500.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_RL1_ABOVE|6|192191|ET192191|E192191|CA Semimonthly|CA
 Tax Reporting Unit|Quebec Labour Standards Taxable|Relationship Tax Unit, Province, Statutory Report
 Type, Statutory Report Code Year to Date|78500.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
```

Example of Quebec Labour Standards: Initializing RL1 Box A Adjustment Balance – Below the Maximum Limit

In this example, you create balance initialization records to initialize an employee's QLS balances, including the RL-1 Box A Adjustment balance, below the maximum limit.

In this example, wages are below the 2020 annual maximum limit of $78,500 and no exempt wages exist. This example illustrates the relationship between QLS balances.

> **Note:** These are only examples and you may require other balances to be initialized depending on your unique requirements. You may need to initialize other balances separately that are part of QLS gross or QLS exempt earnings.

Balance Name	Value	Notes
Gross Earnings	90,000	N/A
RL-1 Box A Adjustment	-20,000	N/A
Quebec Labour Standards Gross	70,000	N/A
Quebec Labour Standards Subject	70,000	There are no Quebec Labour Standard Exempt Wages, so Subject = Gross
Quebec Labour Standards Taxable	70,000	Subject amount of 70,000 is Taxable since it is under the annual limit
Quebec Labour Standards Liability	49.00	Calculated value of Taxable * rate. For example, using the values for 2020: 70,000 * 0.0007 = 49.00

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|QLS_RL1_BELOW|2020/01/01|CA LDG
```

Load the Initialize Balance Batch Lines

Use the InitializeBalanceBatchLine.dat file to create the batch lines.

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
Value|AreaOne|ContextOneName|ContextOneValue|ContextTwoName|ContextTwoValue
MERGE|InitializeBalanceBatchLine|CA LDG|QLS_RL1_BELOW|1|192191|ET192191|E192191|CA Semimonthly|CA Tax
 Reporting Unit|Gross Earnings|Relationship Tax Unit, Province, Statutory Report Type Year to Date|90000.00|
19|Year End Forms|T4_RL1||
MERGE|InitializeBalanceBatchLine|ZHRX_CAVS_ST_LDGMAIN|GK_LS_BI_TESTA|2|192191|ET192191|E192191|CA
 Semimonthly|CA Tax Reporting Unit|RL-1 Box A Adjustment|Relationship Tax Unit Year to Date|-20000.00|19|
Year End Forms|T4_RL1|Statutory Reporting Code|7453627101RS0001
MERGE|InitializeBalanceBatchLine|ZHRX_CAVS_ST_LDGMAIN|GK_LS_BI_TESTA|3|192191|ET192191|E192191|CA
 Semimonthly|CA Tax Reporting Unit|Quebec Labour Standards Gross|Relationship Tax Unit, Province, Statutory
 Report Type, Statutory Report Code Year to Date|70000.00|19|Year End Forms|T4_RL1|Statutory Reporting Code|
7453627101RS0001
```

```
MERGE|InitializeBalanceBatchLine|ZHRX_CAVS_ST_LDGMAIN|GK_LS_BI_TESTA|4|192191|ET192191|E192191|CA
 Semimonthly|CA Tax Reporting Unit|Quebec Labour Standards Subject|Relationship Tax Unit, Province,
 Statutory Report Type, Statutory Report Code Year to Date|70000.00|19|Year End Forms|T4_RL1|Statutory
 Reporting Code|7453627101RS0001
MERGE|InitializeBalanceBatchLine|ZHRX_CAVS_ST_LDGMAIN|GK_LS_BI_TESTA|5|192191|ET192191|E192191|CA
 Semimonthly|CA Tax Reporting Unit|Quebec Labour Standards Taxable|Relationship Tax Unit, Province,
 Statutory Report Type, Statutory Report Code Year to Date|70000.00|19|Year End Forms|T4_RL1|Statutory
 Reporting Code|7453627101RS0001
MERGE|InitializeBalanceBatchLine|ZHRX_CAVS_ST_LDGMAIN|GK_LS_BI_TESTA|6|192191|ET192191|E192191|CA
 Semimonthly|CA Tax Reporting Unit|Quebec Labour Standards Liability|Relationship Tax Unit, Province,
 Statutory Report Type, Statutory Report Code Year to Date|49.00|19|Year End Forms|T4_RL1|Statutory
 Reporting Code|7453627101RS0001
```

Tax Credit Information

Overview of Canadian Tax Credit Information Cards

Tax Credit Information calculation cards store the information required to accurately calculate federal and provincial taxes for an employee.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, Tax Credit Information calculation cards are auto-generated for your Canadian new hires. You then need to:

- Update the Canadian Taxation, Federal Tax, Employment Insurance, and Canada Pension Plan card components, if applicable.
- Create a Provincial Tax card component, if applicable.
- Create a Payroll Tax card component, if applicable.
- Create a Quebec Tax card component, if applicable.
- Create a tax reporting unit association for the Canadian Taxation card component.

If your employees were created in the Oracle Human Capital Management Cloud without a Payroll or Payroll Interface license then you need to create the complete Tax Credit Information calculation card.

Generating the Surrogate ID of the Tax Reporting Unit

The component sequence for the Canadian Taxation card component is the surrogate ID of the employee's tax reporting unit. It is an application-generated value to uniquely identify the tax reporting unit. You may use the Tax Reporting Unit BI Publisher report to extract your tax reporting units and their surrogate ID values. Refer to the Cloud Customer Connect topic Report: *Tax Reporting Unit*.

Example Files

In addition to the example files provided in topics within this section, refer to Cloud Customer Connect's HCM Integration resource sharing center for examples of how to create complete Tax Credit Information calculation cards.

Tax Credit Information Card Record Types

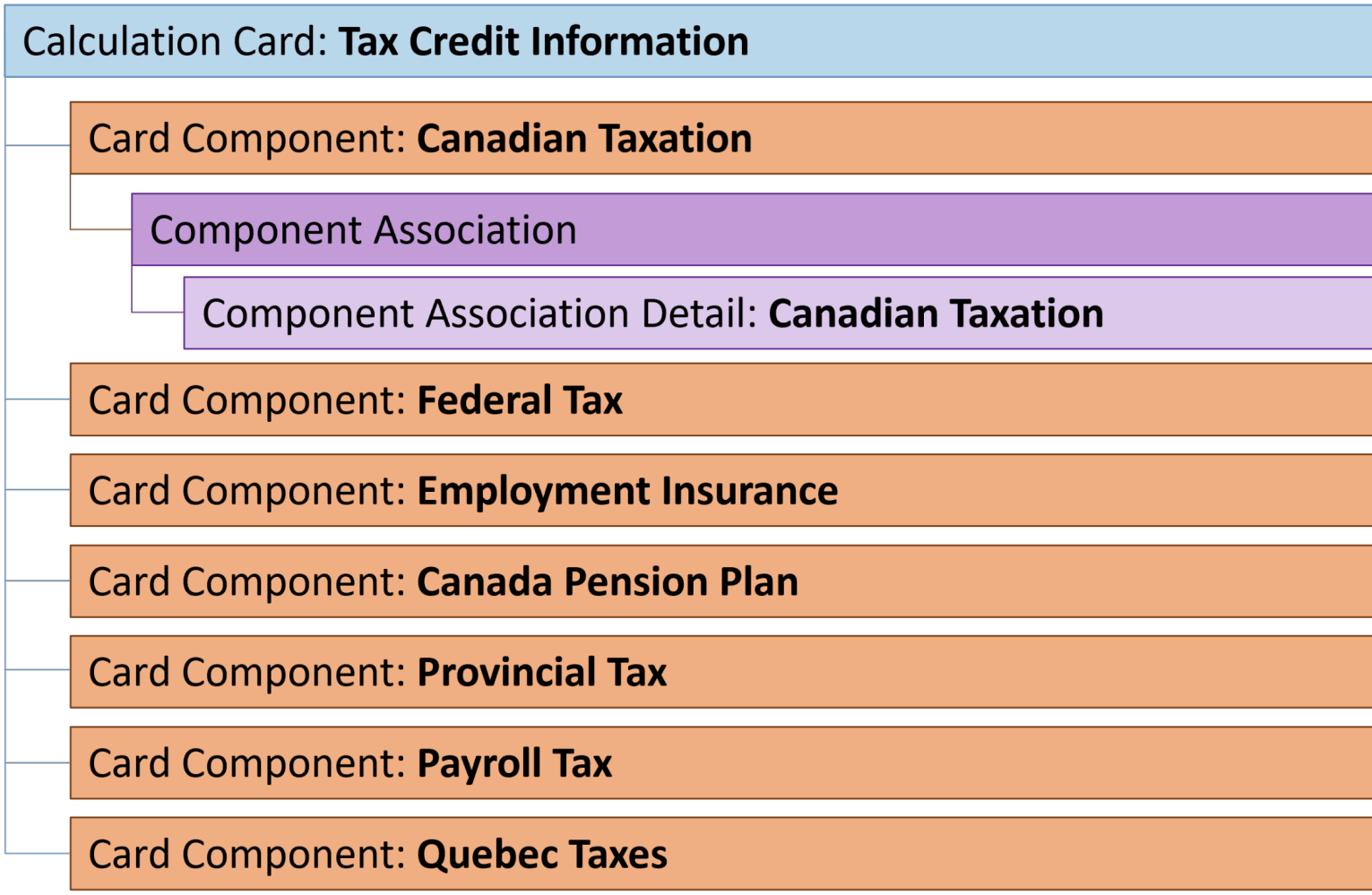
The Tax Credit Information card is bulk-loaded using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various localization requirements.

The Tax Credit Information card utilizes the following Calculation Card record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Supply a component detail record for each flexfield context required by each card component.	ComponentDetail
Calculation Value Definition	Allows the creation of value definitions so that overriding values can be specified on the card component. Details of the specific value definitions are provided in the following sections.	CalculationValueDefinition
Enterable Calculation Value	Used to specify an overriding value for each calculation value definition.	EnterableCalculationValue
Component Association	Associates the Canadian Taxation component with the Tax Reporting Unit the employee reports to.	ComponentAssociation
Component Association Detail	Associates the Canadian Taxation component with the employee's assignments.	ComponentAssociationDetails

Tax Credit Information Calculation Card Hierarchy

The hierarchy of Calculation Card components applicable to Tax Credit Information are described in this diagram:



The card components can have flexfield segments and override values. This section has topics that describe how to provide data for each card component and the component association.

Guidelines for Loading Canadian Tax Credit Information Cards

Supply one Calculation Card record for every Canadian employee you are maintaining Tax Credit Information for.

Even if you are updating an existing calculation card and the calculation card itself is not being updated, still include the calculation card record to group other related data supplied in the file.

Examples of supplying these attributes are supplied in the following topics.

Tax Credit Information Calculation Card Attributes

The Tax Credit Information calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Tax Credit Information calculation card. For new calculation cards supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Tax Credit Information'.
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.

Examples of supplying these attributes are supplied in the following topics:

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Taxation Card Components
- Guidelines for Loading Canadian Federal Tax Card Components
- Guidelines for Loading Canadian Employment Insurance Card Components
- Guidelines for Loading Canada Pension Plan Card Components
- Guidelines for Loading Canadian Provincial Tax Card Components
- Guidelines for Loading Canadian Payroll Tax Card Components
- Guidelines for Loading Quebec Taxes Card Components
- Guidelines for Loading Canadian Taxation Component Associations

Guidelines for Loading Canadian Taxation Card Components

Supply a Canadian Taxation card component and detail values for each tax reporting unit for the employee.

Canadian Taxation Card Component Hierarchy

The Canadian Taxation card component uses the Taxation Context flexfield context and data for this is loaded using the Component Detail record type. The employee's tax reporting unit is associated with the Canadian Taxation card component. This association is described in the *Guidelines for Loading Canadian Taxation Component Associations* topic.

Card Component Attributes for Canadian Taxation

The Canadian Taxation Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Canadian Taxation card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Tax Credit Information calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Canadian Taxation card component. This must be the same as the EffectiveStartDate on the Tax Credit Information calculation card. If updating an existing Canadian Taxation card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
DirCardCompDefName	N/A	The component definition name. Specify 'Canadian Taxation'.
ComponentSequence	N/A	The component sequence for the Canadian Taxation card component is the surrogate ID of the employee's tax reporting unit. It is an application-generated value to uniquely identify the tax reporting unit. Refer to Cloud Customer Connect topic Report: Tax Reporting Units for details on how to extract this information.
Context1	N/A	The surrogate ID of the employee's tax reporting unit. Supply the same value as provided for the ComponentSequence.

Card Component Detail Attributes for Canadian Taxation

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Taxation Context (HRX_CA_WTH_TAXATION)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Canadian Taxation card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Canadian Taxation card component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_CA_WTH_TAXATION'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Taxation Component Associations
- Example of Updating Canadian Taxation Card Components

Example of Updating Canadian Taxation Card Components

The given file lines update an auto-generated Canadian Taxation card component of the Tax Credit Information calculation card.

All records are identified by user keys.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate
MERGE|CalculationCard|CA LDG|Tax Credit Information|E816423|1|2019/05/02

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate|DirCardCompDefName|ComponentSequence|Context1
MERGE|CardComponent|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Canadian Taxation|300100044131319|
300100044131319

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer
 DF|provinceOfEmployement(Deduction Developer DF=HRX_CA_WTH_TAXATION)|oraHrxCaVlPlan(Deduction Developer
 DF=HRX_CA_WTH_TAXATION)|oraHrxCaVacServDate(Deduction Developer DF=HRX_CA_WTH_TAXATION)
MERGE|ComponentDetail|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Canadian Taxation|300100044131319|
HRX_CA_WTH_TAXATION|HRX_CA_WTH_TAXATION|19||
```

Related Topics

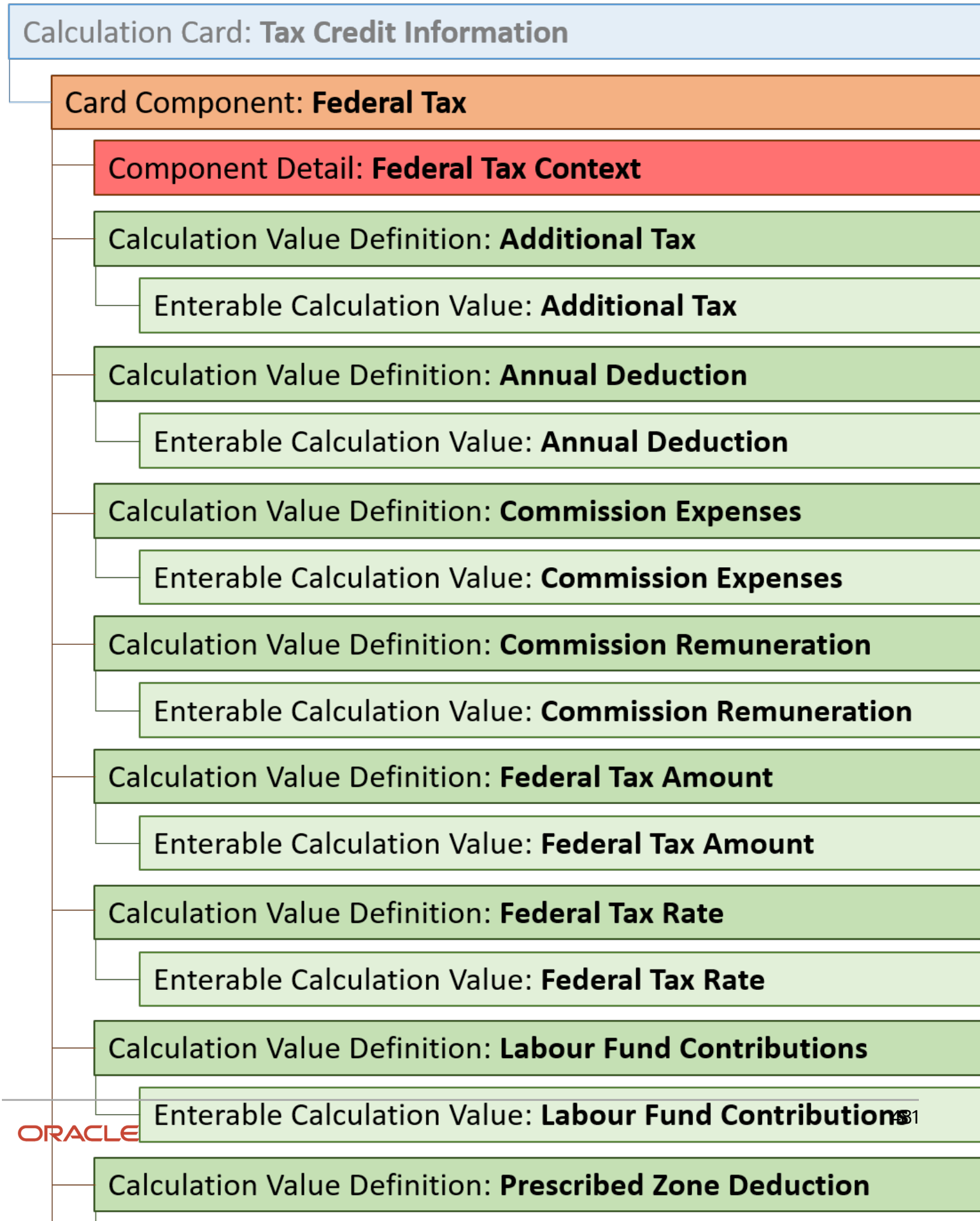
- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Taxation Card Components

Guidelines for Loading Canadian Federal Tax Card Components

The Federal Tax card component is used to capture information that will impact Federal Tax calculations for the employee.

Each Tax Credit Information calculation card should have one Federal Tax card component.

Federal Tax Card Component Hierarchy


Calculation Card: Tax Credit Information
Card Component: Federal Tax
Component Detail: Federal Tax Context
Calculation Value Definition: Additional Tax
Enterable Calculation Value: Additional Tax
Calculation Value Definition: Annual Deduction
Enterable Calculation Value: Annual Deduction
Calculation Value Definition: Commission Expenses
Enterable Calculation Value: Commission Expenses
Calculation Value Definition: Commission Remuneration
Enterable Calculation Value: Commission Remuneration
Calculation Value Definition: Federal Tax Amount
Enterable Calculation Value: Federal Tax Amount
Calculation Value Definition: Federal Tax Rate
Enterable Calculation Value: Federal Tax Rate
Calculation Value Definition: Labour Fund Contributions
Enterable Calculation Value: Labour Fund Contributions
Calculation Value Definition: Prescribed Zone Deduction

The Federal Tax card component uses the Federal Tax Context flexfield context and data for this is loaded using the Component Detail record type. The Federal Tax card component has multiple value definitions, override values are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Card Component Attributes for Federal Tax

The Federal Tax Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Federal Tax card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Tax Credit Information calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the Federal Tax card component. This must be the same as the EffectiveStartDate on the Tax Credit Information calculation card. If updating an existing Federal Tax card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Federal Tax'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.

Card Component Detail Attributes for Federal Tax

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Federal Tax Context (HRX_CA_WTH_FEDERAL_INCOME_TAX)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Federal Tax card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the **Federal Tax** card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_CA_WTH_FEDERAL_INCOME_TAX'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Federal Tax Value Definitions

The Federal Tax card component uses these value definitions to supply override values:

Value Definition Name	Functional Description
Total Claim Amount	The federal tax exemption amount.

Value Definition Name	Functional Description
	Note: This value defaults to the basic amount populated from the Load Payroll Tax Information process. For employees with the basic amount, do not populate the Total Claim Amount, as the value will be read as an override and will not be updated by the Load Payroll Tax Information process when the basic amounts change. If the employee has a total claim amount different than the basic amount, the amount should be entered here. This override is not reset to the basic amount or updated at the beginning of the year. It will remain the same until modified by the user, or reset using the Tax Overrides Report.
Additional Tax	The amount an employee elects to have deducted as extra tax, in addition to the tax calculation. The amount is entered in dollars.
Federal Tax Rate	The federal income tax rate.
Federal Tax Amount	The federal income tax amount. **Note:** If an Additional Tax value is entered for the employee, it is deducted in addition to the Federal Tax Amount.
Federal Lump Sum Rate	An override to the rate used for calculation of taxes for lump sum payments. If you do not override the rate, the deduction calculation uses the prescribed rate.
Annual Deduction	The tax reduction relating to child care and alimony expenses that is authorized by a tax services office.
Labour Fund Contributions	A tax calculation reduction applicable only when the employee has deductions related to labour sponsored funds.
Prescribed Zone Deduction	A federal tax calculation reduction for those employees that live in a prescribed zone.
Other Tax Credits	An additional tax credit used to reduce the income tax amount, as authorized by the government.
RRSP Limit	The RRSP contribution for the year is capped at an annual limit. This includes both the employee contribution as well as the matched employer contribution taxable benefit This field defaults to the published RRSP annual limit populated from the Load Payroll Tax Information process. However if required, you may override the annual limit for the employee.
Annual Income Including Commission	The amount of income a commissioned employee is expected to earn in the calendar year.
Commission Expenses	The amount of expenses a commissioned employee is expected to have in the calendar year.

Note: The names supplied here match those displayed on the user interface. The same value is supplied to the ValueDefinitionName attribute on the calculation value definition record when you're providing an overview value.

Calculation Value Definition Attributes for Federal Tax

The Calculation Value Definition record type specifies the value definition name for the override value.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName,	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, DirCardCompDefName	calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Federal Tax card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent Federal Tax card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent Federal Tax card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent Federal Tax card component and a CalculationCard record for the owning Tax Credit Information card.

Enterable Calculation Value Attributes for Federal Tax

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName,	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record. **Note:** Supply -999999999 to indicate a null (blank) value.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Example of Updating Canadian Federal Tax Card Components

Example of Updating Canadian Federal Tax Card Components

The given file lines update the auto-generated Federal Tax card component of the Tax Credit Information calculation card.

This example includes override records for all value definitions supported by the Federal Tax card component.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate
MERGE|CalculationCard|CA LDG|Tax Credit Information|E816423|1|2019/05/02

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
federalTaxExempt(Deduction Developer DF=HRX_CA_WTH_FEDERAL_INCOME_TAX)
MERGE|ComponentDetail|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|
HRX_CA_WTH_FEDERAL_INCOME_TAX|HRX_CA_WTH_FEDERAL_INCOME_TAX|Y
```

```
METADATA|CalculationValueDefinition|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|
CardSequence|EffectiveStartDate|DirCardCompDefName|ComponentSequence|ValueDefinitionName
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Total
 Claim Amount
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Additional
 Tax
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Federal
 Tax Rate
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Federal
 Tax Amount
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Federal
 Lump Sum Rate
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Annual
 Deduction
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Labour
 Fund Contributions
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Prescribed
 Zone Deduction
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Other Tax
 Credits
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|RRSP Limit
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Annual
 Income Including Commission
MERGE|CalculationValueDefinition|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Commission
 Expenses

METADATA|EnterableCalculationValue|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|
CardSequence|EffectiveStartDate|DirCardCompDefName|ComponentSequence|ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Total Claim
 Amount|100
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Additional
 Tax|100
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Federal Tax
 Rate|100
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Federal Tax
 Amount|100
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Federal
 Lump Sum Rate|11
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Annual
 Deduction|100
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Labour Fund
 Contributions|100
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Prescribed
 Zone Deduction|100
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Other Tax
 Credits|100
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|RRSP Limit|
100
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Annual
 Income Including Commission|100
MERGE|EnterableCalculationValue|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Federal Tax|1|Commission
 Expenses|100
```

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Federal Tax Card Components

Guidelines for Loading Canadian Employment Insurance Card Components

The Employment Insurance card component is used to capture information that will impact Employment Insurance calculations for the employee.

Each Tax Credit Information calculation card should have one Employment Insurance card component.

Employment Insurance Card Component Hierarchy

The Employment Insurance card component utilizes the Employment Insurance Context flexfield context and data for this is loaded using the Component Detail record type. The following sections describe how to supply valid file lines for these record types.

Card Component Attributes for Employment Insurance

The Employment Insurance Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employment Insurance card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Tax Credit Information calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The start date of the Employment Insurance card component. This must be the same as the EffectiveStartDate on the Tax Credit Information calculation card. If updating an existing. Employment Insurance card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Employment Insurance'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.

Card Component Detail Attributes for Employment Insurance

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Employment Insurance Context (HRX_CA_WTH_FEDERAL_EMPLOYMENT_INSURANCE)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **Employment Insurance** card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the **Employment Insurance** card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_CA_WTH_FEDERAL_EMPLOYMENT_INSURANCE'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Example of Updating Canadian Employment Insurance Card Components

Example of Updating Canadian Employment Insurance Card Components

The given file lines update the auto-generated Employment Insurance card component of the Tax Credit Information calculation card.

The Employment Insurance card component populates the EI Exempt checkbox seen in the Federal section of the user interface.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate
MERGE|CalculationCard|CA LDG|Tax Credit Information|E816423|1|2019/05/02

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Employment Insurance|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
eiExempt(Deduction Developer DF=HRX_CA_WTH_FEDERAL_EMPLOYMENT_INSURANCE)
MERGE|ComponentDetail|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Employment Insurance|1|
HRX_CA_WTH_FEDERAL_EMPLOYMENT_INSURANCE|HRX_CA_WTH_FEDERAL_EMPLOYMENT_INSURANCE|Y
```

Related Topics

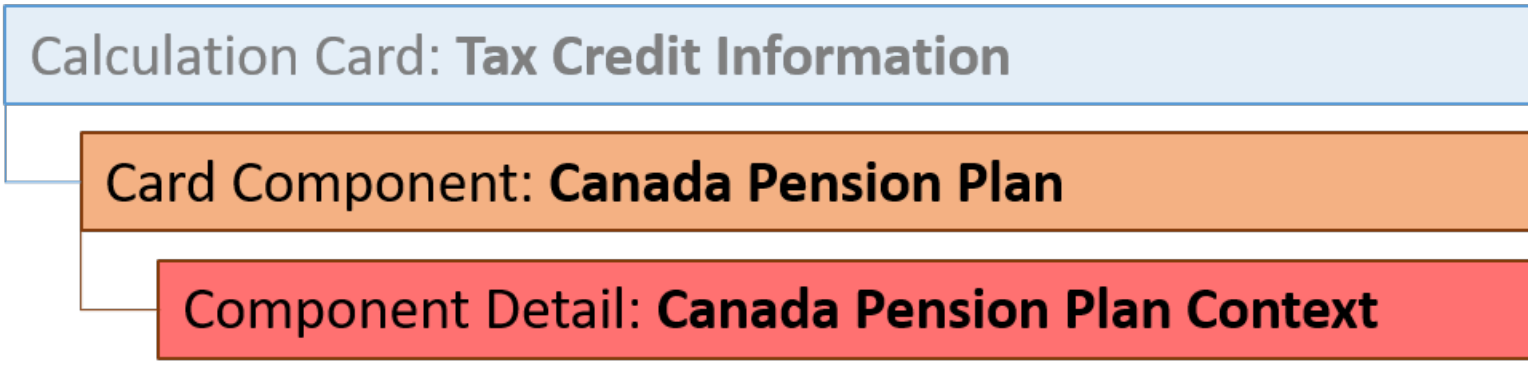
- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Employment Insurance Card Components

Guidelines for Loading Canada Pension Plan Card Components

The Canada Pension Plan (CPP) card component is used to capture information that will impact Canada Pension Plan calculations for the employee.

Each Tax Credit Information calculation card should have one Canada Pension Plan card component.

Canada Pension Plan Card Component Hierarchy



The Canada Pension Plan card component utilizes the Canada Pension Plan Context flexfield context and data for this is loaded using the Component Detail record type. The following sections describe how to supply valid file lines for these record types.

Card Component Attributes for Canada Pension Plan

The Canada Pension Plan Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Canada Pension Plan card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Tax Credit Information calculation card should be identified by using the same key type used to identify the calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the Canada Pension Plan card component. This must be the same as the EffectiveStartDate on the Tax Credit Information calculation card. If updating an existing Canada Pension Plan card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Canada Pension Plan'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.

Card Component Detail Attributes for Canada Pension Plan

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Canada Pension Plan Context (HRX_CA_WTH_FEDERAL_PENSION_PLAN)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Canada Pension Plan card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Canada Pension Plan card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_CA_WTH_FEDERAL_PENSION_PLAN'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_CA_WTH_FEDERAL_PENSION_PLAN'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards

Example of Updating Canadian Pension Plan Card Components

These file lines update the auto-generated Canada Pension Plan card component of the Tax Credit Information calculation card.

The Canada Pension Plan card component populates the CPP Exempt checkbox and dates seen in the Federal section of the user interface.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate
MERGE|CalculationCard|CA LDG|Tax Credit Information|E816423|1|2019/05/02

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Canada Pension Plan|1|
```

```
METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|CardSequence|
EffectiveStartDate|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer
 DF|cppExempt(Deduction Developer DF=HRX_CA_WTH_FEDERAL_PENSION_PLAN)|cppRevocationDate(Deduction Developer
 DF=HRX_CA_WTH_FEDERAL_PENSION_PLAN)|cppElectionDate(Deduction Developer DF=HRX_CA_WTH_FEDERAL_PENSION_PLAN)
MERGE|ComponentDetail|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Canada Pension Plan|1|
HRX_CA_WTH_FEDERAL_PENSION_PLAN|HRX_CA_WTH_FEDERAL_PENSION_PLAN|Y||
```

Related Topics

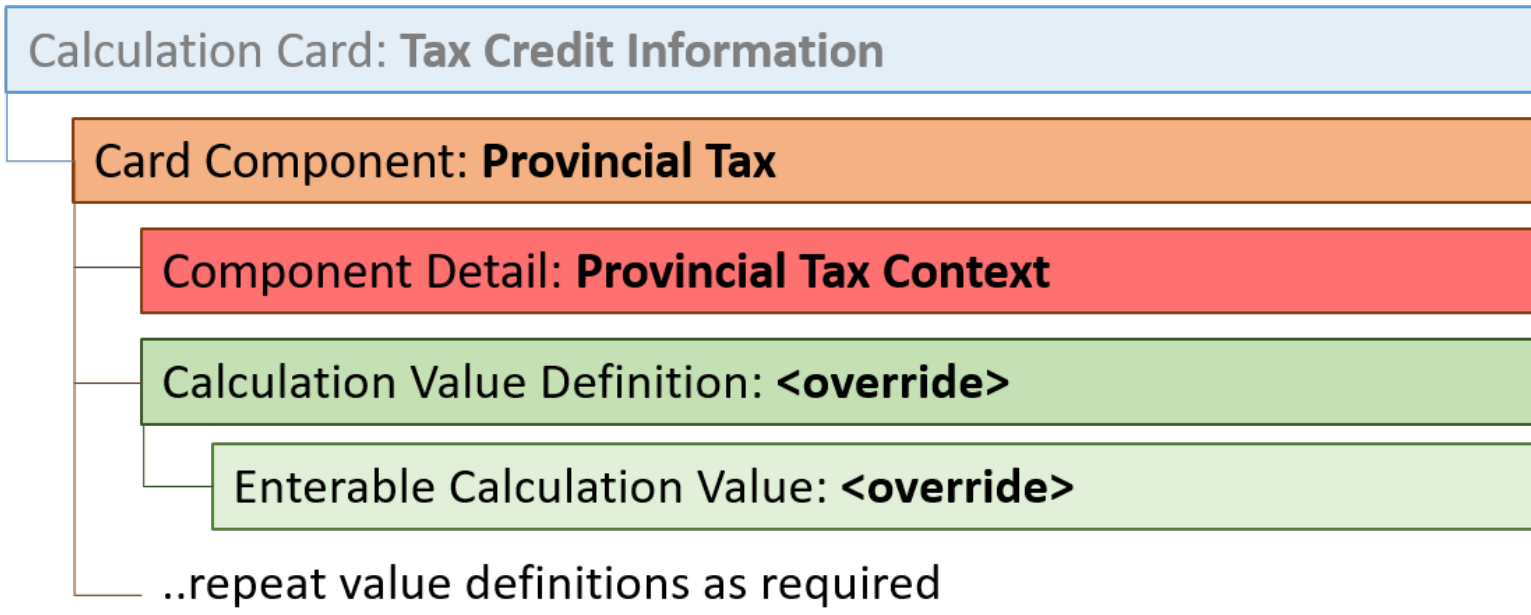
- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Example of Updating Canadian Pension Plan Card Components

Guidelines for Loading Canadian Provincial Tax Card Components

The Provincial Tax card component is used to capture information that will impact provincial tax calculations for the employee.

Each Tax Credit Information calculation card should have one Provincial Tax card component.

Provincial Tax Card Component Hierarchy



All provinces use the Provincial Tax Context flexfield context and data for this is supplied using the Component Detail record type. All provinces use value definitions and these are supplied using the Calculation Value Definition and Enterable Calculation Value record types. The value definitions differ across the provinces. Refer to the Value Definitions table below.

Card Component Attributes for Provincial Tax

The Provincial Tax Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the **Provincial Tax** card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Tax Credit Information calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the Provincial Tax card component. This must be the same as the EffectiveStartDate on the Tax Credit Information calculation card. If updating an existing Provincial Tax card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Provincial Tax'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.
Context1	N/A	The code of the province. Refer to the *Geography Codes for Canadian Provinces* topic.

Card Component Detail Attributes for Provincial Tax

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Provincial Tax Context (HRX_CA_WTH_PROVINCE_INCOME_TAX)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **Provincial Tax** card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the **Provincial Tax** card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_CA_WTH_PROVINCE_INCOME_TAX'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_CA_WTH_PROVINCE_INCOME_TAX'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Provincial Tax Value Definitions

The Provincial Tax card component uses value definitions to supply override values. The province determines which value definitions are available, as not all values are applicable to all provinces

These value definitions are applicable to all provinces except:

- Northwest Territories (NT)
- Nunavut (NU)

For the value definitions available for NT and NU, see Guidelines for Loading Canadian Payroll Tax Card Components.

Value Definition Name	Functional Description
Other Tax Credits	An additional tax credit used to reduce the income tax amount, as authorized by the government.
Total Claim Amount	This value represents the provincial tax exemption amount. **Note:** This value defaults to the basic amount populated from the Load Payroll Tax Information process. For employees with the basic amount, do not populate the Total Claim Amount, as the value will be read as an override and will not be updated by the Load Payroll Tax Information process when the basic amounts change. If the employee has a total claim amount different than the basic amount, the amount should be entered here. This override is not reset to the basic amount or updated at the beginning of the year. It will remain the same until modified by the user, or reset using the Tax Overrides Report.

This value definition is applicable to the following provinces:

- British Columbia (BC)
- Manitoba (MB)
- New Brunswick (NB)
- Newfoundland and Labrador (NL)
- Nova Scotia (NS)
- Quebec (QC)
- Saskatchewan (SK)
- Yukon (YT)

Value Definition Name	Functional Description
Labour Fund Contributions	A tax calculation reduction applicable only when the employee has deductions related to labour-sponsored funds.

The following value definitions are only applicable to Quebec (QC):

Value Definition Name	Functional Description
Additional Tax	The amount an employee elects to have deducted as extra tax, in addition to the tax calculation. The amount is entered in dollars.
Annual Deduction	Tax reduction relating to child care and alimony expenses that is authorized by a tax services office.
Commission Expenses	The amount of expenses a commissioned employee is expected to have in the calendar year.
Annual Commission Income	The amount of income a commissioned employee is expected to earn in the calendar year.
Designated Remote Area Deduction	A provincial tax calculation reduction for those employees that live in a designated remote area.
Provincial Tax Amount	This is an override for the provincial income tax amount.
Provincial Tax Rate	This is an override for provincial income tax rate.
Provincial Lump Sum Rate	This is an override to the provincial rate used for calculation of taxes for lump sum payments. If you do not override the rate, the deduction calculation uses the prescribed rate.

Calculation Value Definition Attributes for Provincial Tax

The Calculation Value Definition record type specifies the value definition name for the override value.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **Provincial Tax** card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent **Provincial Tax** card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent **Provincial Tax** card component. Specify the same value as provided on the parent card component record.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.
Context1	N/A	The code of the province the provincial tax override is for.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent Provincial Tax card component and a CalculationCard record for the owning Tax Credit Information card.

Enterable Calculation Value Attributes for Provincial Tax

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Federal Tax Card Components
- Example of Loading Canadian Provincial Tax Card Components

Example of Loading Canadian Provincial Tax Card Components

The given file lines create the Provincial Tax card component for an auto-generated Tax Credit Information calculation card. This example is for British Columbia.

The information provided on the Provincial Tax card component is displayed on the Regional section of the user interface.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
AssignmentNumber|CardSequence|EffectiveStartDate
MERGE|CalculationCard|||CA LDG|Tax Credit Information|E816423|1|2019/05/02

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
AssignmentNumber|CardSequence|EffectiveStartDate|DirCardCompDefName|ComponentSequence|Context1
MERGE|CardComponent|VISION|TCI_816423_PT_3|CA LDG|Tax Credit Information|E816423|1|2019/05/02|Provincial
 Tax|3|3

METADATA|ComponentDetail|SourceSystemId|SourceSystemOwner|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|provinceTaxExempt(Deduction Developer DF=HRX_CA_WTH_PROVINCE_INCOME_TAX)|
numberOfDependentsUnderAge(Deduction Developer DF=HRX_CA_WTH_PROVINCE_INCOME_TAX)|
numberOfDisabledDependents(Deduction Developer DF=HRX_CA_WTH_PROVINCE_INCOME_TAX)
MERGE|ComponentDetail|TCI_816423_PT_3_PIT|VISION|CA LDG|TCI_816423_PT_3|2019/05/02|Provincial Tax|
HRX_CA_WTH_PROVINCE_INCOME_TAX|HRX_CA_WTH_PROVINCE_INCOME_TAX|Y||

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName|Context1
MERGE|CalculationValueDefinition|VISION|TCI_816423_PT_3_VD1|CA LDG|TCI_816423_PT_3|2019/05/02|Provincial
 Tax|Total Claim Amount|3
MERGE|CalculationValueDefinition|VISION|TCI_816423_PT_3_VD2|CA LDG|TCI_816423_PT_3|2019/05/02|Provincial
 Tax|Other Tax Credits|3
MERGE|CalculationValueDefinition|VISION|TCI_816423_PT_3_VD3|CA LDG|TCI_816423_PT_3|2019/05/02|Provincial
 Tax|Labour Fund Contributions|3

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName|Context1|Value1
MERGE|EnterableCalculationValue|VISION|TCI_816423_PT_3_VD1|CA LDG|TCI_816423_PT_3_VD1|2019/05/02|Provincial
 Tax|Total Claim Amount|3|100
MERGE|EnterableCalculationValue|VISION|TCI_816423_PT_3_VD2|CA LDG|TCI_816423_PT_3_VD2|2019/05/02|Provincial
 Tax|Other Tax Credits|3|100
MERGE|EnterableCalculationValue|VISION|TCI_816423_PT_3_VD3|CA LDG|TCI_816423_PT_3_VD3|2019/05/02|Provincial
 Tax|Labour Fund Contributions|3|100
```

Related Topics

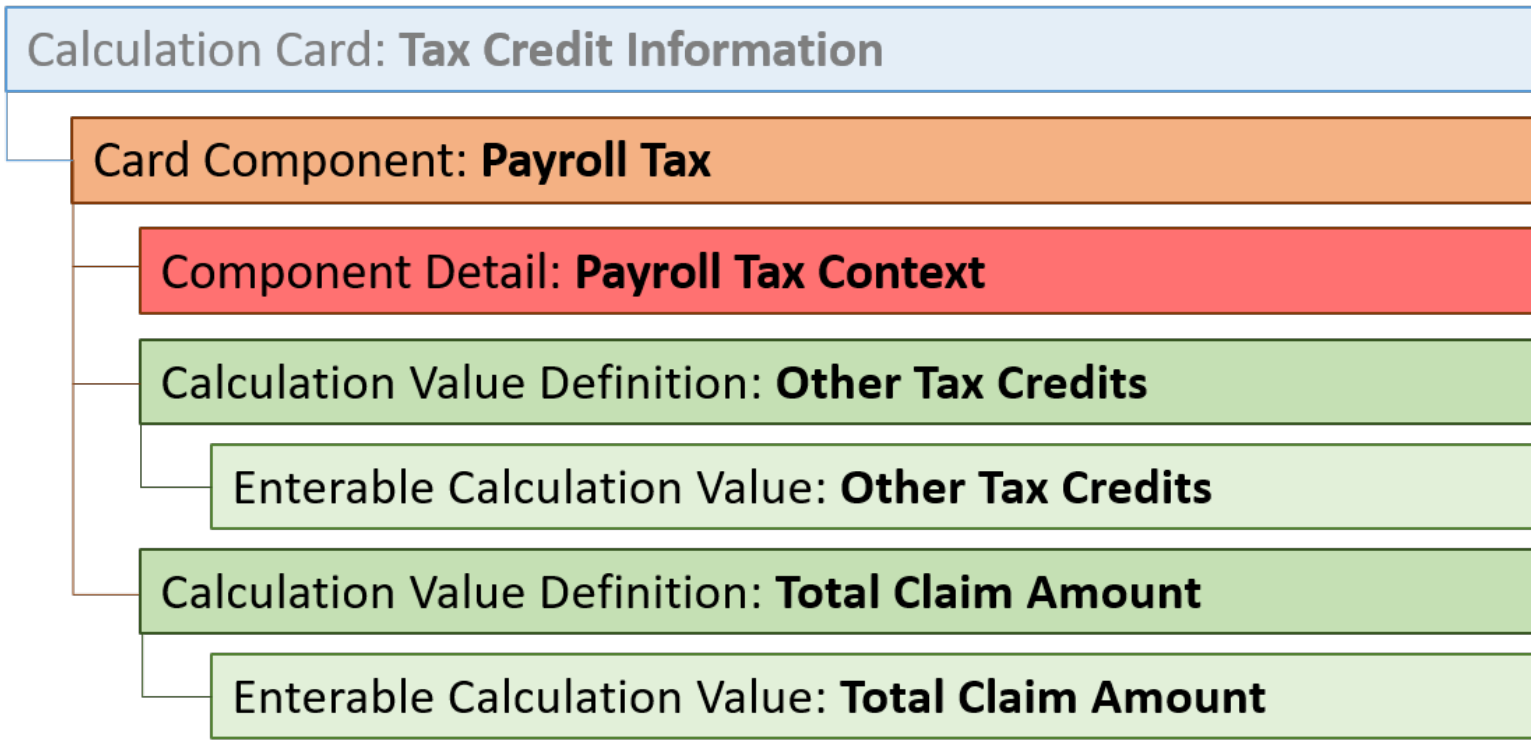
- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Provincial Tax Card Components

Guidelines for Loading Canadian Payroll Tax Card Components

The Payroll Tax card component is used to capture information that will impact an employee's payroll tax calculations for the provinces of Northwest Territories and Nunavut.

Each Tax Credit Information calculation card should have one Payroll Tax card component, if the province is Northwest Territories and Nunavut.

Payroll Tax Card Component Hierarchy



Most provinces use the Provincial Tax card component, but Northwest Territories and Nunavut use the Payroll Tax card component.

Card Component Attributes for Payroll Tax

The Payroll Tax Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Payroll Tax card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Tax Credit Information calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the Payroll Tax card component. This must be the same as the EffectiveStartDate on the Tax Credit Information calculation card. If updating an existing Payroll Tax card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Payroll Tax'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.
Context1	N/A	The code of the province. Refer to the Geography Code for Canadian Provinces topic.

Card Component Detail Attributes for Payroll Tax

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Payroll Tax Context (HRX_CA_WTH_PROVINCE_PAYROLL_TAX)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **Payroll Tax** card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the **Payroll Tax** card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_CA_WTH_PROVINCE_PAYROLL_TAX'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Payroll Tax Value Definitions

The Payroll Tax card component uses these value definitions to supply override values:

Value Definition Name	Functional Description
Total Claim Amount	The provincial tax exemption amount. **Note:** This value defaults to the basic amount populated from the Load Payroll Tax Information process. For employees with the basic amount, do not populate the Total Claim Amount, as the value will be read as an override and will not be updated by the Load Payroll Tax Information process when the basic amounts change. If the employee has a total claim amount different than the basic amount, the amount should be entered here. This override is not reset to the basic amount or updated at the beginning of the year. It will remain the same until modified by the user, or reset using the Tax Overrides Report.
Other Tax Credits	An additional tax credit used to reduce the income tax amount, as authorized by the government.

Calculation Value Definition Attributes for Payroll Tax

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Payroll Tax card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent Payroll Tax card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent Payroll Tax card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent Payroll Tax card component and a CalculationCard record for the owning Tax Credit Information card.

Enterable Calculation Value Attributes for Payroll Tax

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Example of Loading Canadian Payroll Tax Card Components

Example of Loading Canadian Payroll Tax Card Components

The given file lines create the Payroll Tax card component for an auto-generated Tax Credit Information calculation card.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber|CardSequence
MERGE|CalculationCard|||CA LDG|Tax Credit Information|2019/05/02|E816423|1

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber|CardSequence|DirCardCompDefName|ComponentSequence|Context1
MERGE|CardComponent|VISION|TCI_816423_PY_11|CA LDG|Tax Credit Information|2019/05/02|E816423|1|Payroll Tax|
11|11

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|FLEX:Deduction
 Developer DF|payrollTaxExempt(Deduction Developer DF=HRX_CA_WTH_PROVINCE_PAYROLL_TAX)
MERGE|ComponentDetail|VISION|TCI_816423_PY_11_PAY|CA LDG|TCI_816423_PY_11|2019/05/02|Payroll Tax|
HRX_CA_WTH_PROVINCE_PAYROLL_TAX|HRX_CA_WTH_PROVINCE_PAYROLL_TAX|Y
```

Related Topics

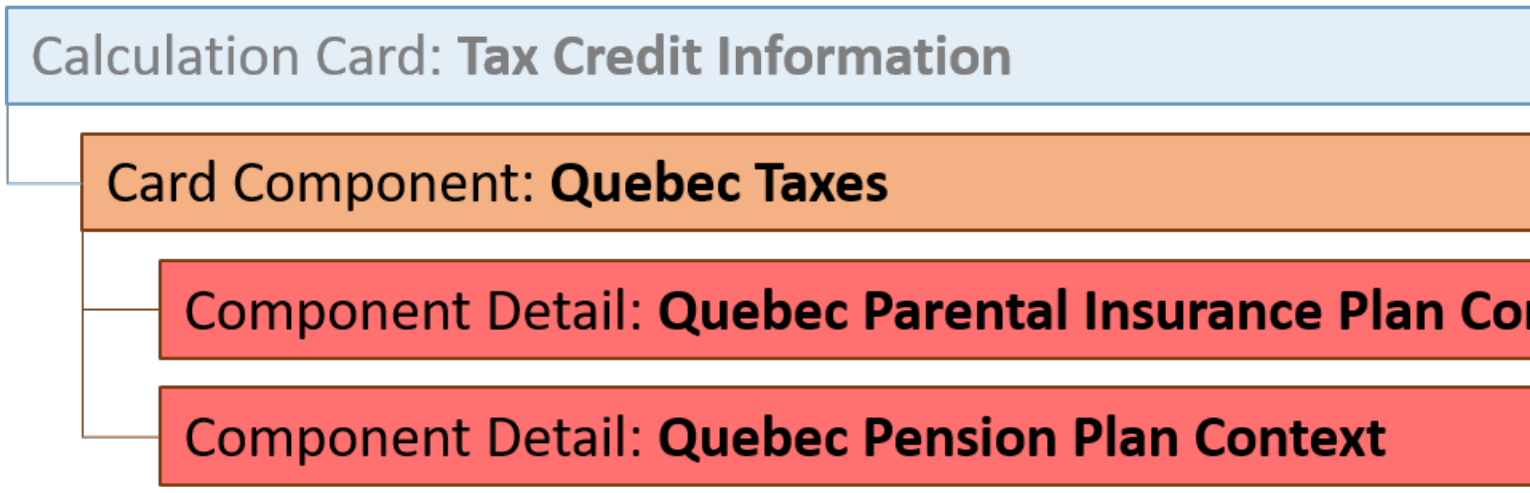
- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Payroll Tax Card Components

Guidelines for Loading Quebec Taxes Card Components

The Quebec Taxes card component is used to capture information that will impact Quebec Pension Plan and Quebec Parental Insurance Plan calculations for the employee.

Each Tax Credit Information calculation card should have one Quebec Taxes card component, if the employee works in Quebec.

Quebec Taxes Card Component Hierarchy



The Quebec Taxes card component utilizes two flexfield contexts and data for these are loaded using the Component Detail record type. The following sections describe how to supply valid file lines for these record types.

Card Component Attributes for Quebec Taxes

The Quebec Taxes Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Quebec Taxes card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Tax Credit Information calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the Quebec Taxes card component. This must be the same as the EffectiveStartDate on the Tax Credit Information calculation card.If updating an existing Quebec Taxes card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Specify 'Quebec Taxes'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.
Context1	N/A	The code of the province. Specify '19'.

Card Component Detail Attributes for Quebec Taxes

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield contexts:

- Quebec Parental Insurance Plan Context (HRX_CA_WTH_PROVINCE_PARENTAL_INSURANCE)
- Quebec Pension Plan Context (HRX_CA_WTH_PROVINCE_PENSION_PLAN)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Quebec Taxes card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Quebec Taxes card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Quebec Taxes card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_CA_WTH_PROVINCE_PARENTAL_INSURANCE'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Example of Loading Quebec Taxes Card Components

Example of Loading Quebec Taxes Card Components

The given file lines create the Quebec Taxes card component of an auto-generated Tax Credit Information calculation card.

> **Note:** You can maintain Quebec Taxes individually but when creating them they are supplied with the Provincial Tax card component for Quebec.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber|CardSequence
MERGE|CalculationCard|||CA LDG|Tax Credit Information|2019/05/02|E816423|1

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber|CardSequence|DirCardCompDefName|ComponentSequence|Context1
MERGE|CardComponent|VISION|TCI_816423_QB|CA LDG|Tax Credit Information|2019/05/02|E816423|1|Quebec Taxes|19|
19

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|FLEX:Deduction
 Developer DF|ppipExempt(Deduction Developer DF=HRX_CA_WTH_PROVINCE_PARENTAL_INSURANCE)|pppExempt(Deduction
 Developer DF=HRX_CA_WTH_PROVINCE_PENSION_PLAN)
MERGE|ComponentDetail|VISION|TCI_816423_QB_PAR|CA LDG|TCI_816423_QB|2019/05/02|Quebec Taxes|
HRX_CA_WTH_PROVINCE_PARENTAL_INSURANCE|HRX_CA_WTH_PROVINCE_PARENTAL_INSURANCE|Y|
MERGE|ComponentDetail|VISION|TCI_816423_QB_PEN|CA LDG|TCI_816423_QB|2019/05/02|Quebec Taxes|
HRX_CA_WTH_PROVINCE_PENSION_PLAN|HRX_CA_WTH_PROVINCE_PENSION_PLAN||Y
```

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Guidelines for Loading Quebec Taxes Card Components

Guidelines for Loading Canadian Taxation Component Associations

The component association record type associates the Tax Credit Information calculation card with a tax reporting unit.

Tax Credit Information Candian Taxes Association

Calculation Card: **Tax Credit Information**

Card Component: **Canadian Taxation**

Component Association

Component Association Detail: **Canadian Taxation**

Component Association Attributes for Canadian Taxations

The associated tax reporting unit is defined in the component association.

The Component Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName, ComponentSequence	A unique identifier for the Canadian Taxation card component association. For new card component associations supply the source key attributes. You can also identify card components associations with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Canadian Taxationes card component should be identified by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card component association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Tax Credit Information calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Canadian Taxation card component. If source keys are used to identify the card association, you must still specify the

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		LegislativeDataGroupName attribute to identify the tax reporting unit.

Component Association Detail Attributes for Canadian Taxation

The component association details record associates the Canadian Taxation card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Component Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	DirCardDefinitionName, AssignmentNumber, LegislativeDataGroupName, CardSequence, TaxReportingUnitName, DirCardCompDefName, ComponentSequence, AssociationAssignmentNumber	A unique identifier for the component association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, ComponentSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	The parent component association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Taxation Card Components
- Example of Loading Canadian Taxes Component Associations

Example of Loading Canadian Taxes Component Associations

The given file lines create the Component Association for an auto-generated Tax Credit Information calculation card.

User keys are used to identify the Tax Credit Information calculation card and the Canadian Taxation card component referenced in the component association detail record.

```
METADATA|ComponentAssociation|EffectiveStartDate|LegislativeDataGroupName|TaxReportingUnitName|
DirCardDefinitionName|AssignmentNumber|CardSequence|DirCardCompDefName|ComponentSequence
MERGE|ComponentAssociation|2019/05/02|CA LDG|CA Tax Reporting Unit|Tax Credit Information|E816423|1|Canadian
 Taxation|300100044131319

METADATA|ComponentAssociationDetail|EffectiveStartDate|LegislativeDataGroupName|TaxReportingUnitName|
DirCardDefinitionName|AssignmentNumber|CardSequence|DirCardCompDefName|ComponentSequence|
AssociationAssignmentNumber
MERGE|ComponentAssociationDetail|2019/05/02|CA LDG|CA Tax Reporting Unit|Tax Credit Information|E816423|1|
Canadian Taxation|300100044131319|E816423
```

Related Topics

- Overview of Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Tax Credit Information Cards
- Guidelines for Loading Canadian Taxation Card Components
- Guidelines for Loading Canadian Taxation Component Associations

23 Loading Payroll Localization Data for the United Kingdom

Balance Adjustments

Guidelines for Adjusting Balances for the UK

Follow these guidelines for adjusting balances for the UK.

Refer to the **Initializing and Adjusting Balances** chapter for guidance on adjusting balances.

There are a few points to note when adjusting balances for the UK:

- When adjusting balances that have a context, you need to provide this in the batch lines. Some delivered elements have a unique reference, for example Benefits in Kind and Court Orders. In such cases, the reference must also be provided.
- Third Party Payee information might also be needed where this information is required by the element.
- The following elements are available for making adjustments to UK Tax and National Insurance balances:
  - PAYE Adjustment
  - NI Employee Adjustment
  - NI Employer Adjustment
- When Court Orders and Student Loans elements are created for the specific court order or loan, the following elements are also created:
  - <Court Order element name> Adjustment
  - <Court Order element name> Deduction Results
  - <Court Order element Name> Fee Results (where appropriate)

  > **Note:** Use the Deduction Results element to make balance adjustments. The Adjustment element is adjusting the Court Order amount within a payroll run.

- Adjustments to absence balances should use the Balance Adjustment functionality. The use of HCM Data Loader is not recommended for absence balance adjustments. Refer to the document **Setting Up and Migrating UK Statutory Absences (ID 2235239.1)**

Example of Adjusting Earning Element Balances for the UK

This balance adjustment example creates a batch header and group record for an employee to adjust the following balances using an earnings element.

The element input values will adjust the following balances:

- Gross pay and earnings

- Taxable pay
- NIable earnings
- Pensionable pay
- Attachable earnings

Note: When adjusting balances, you might also need to adjust any related balances, depending on your business needs. This is simply an example to illustrate the process.

Load the Balance Adjustment Header and Group

Use the BalanceAdjustmentHeader.dat file to create the batch header and group records.

```
METADATA|BalanceAdjustmentHeader|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|UK LDG|Salary Adjustment

METADATA|BalanceAdjustmentGroup|LegislativeDataGroupName|BatchName|EffectiveDate|PayrollName|
ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|UK LDG|Salary Adjustment|2022/04/06|UK Engineering Monthly||Y|N
```

Load the Balance Adjustment Line and Value

Balance Adjustment Line Data

Line Sequence	1
Batch Name	Salary Adjustment
Payroll Name	UK Engineering Monthly
Consolidation Set Name	N/A
Effective Date	2022/04/06
Assignment Number	E955160008184963
Element Name	UK Salary Payment
Tax Reporting Unit Name	UK Engineering Ltd.

Note: The line records specify the element for which you will supply the values for in the value record.

Balance Adjustment Value Data

Line Sequence	Input Value name	Entry Value
1	Amount to Pay	1200
2	Reporting Unit	None
3	Periodicity	Calendar Month
4	Periodicity Conversion Rule	Standard Rate Annualized

For a balance adjustment value to be correctly associated with its parent balance adjustment line, these attributes on both lines must have identical values:

- Batch Name
- Payroll Name
- Consolidation Set Name
- Effective Date
- Element Name

Use the BalanceAdjusmentLine.dat file to create batch line and value records:

```
METADATA|BalanceAdjustmentLine|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|ElementName|AssignmentNumber|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|UK LDG|Salary Adjustment|UK Engineering Monthly||2022/04/06|1|UK Salary Payment|
E8184963|UK Engineering Ltd
METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|ElementName|InputValueName|EntryValue
MERGE|BalanceAdjustmentValue|UK LDG|Salary Adjustment|UK Engineering Monthly||2022/04/06|1|UK Salary
 Payment|Amount to Pay|1200
MERGE|BalanceAdjustmentValue|UK LDG|Salary Adjustment|UK Engineering Monthly||2022/04/06|2|UK Salary
 Payment|Reporting Unit|None
MERGE|BalanceAdjustmentValue|UK LDG|Salary Adjustment|UK Engineering Monthly||2022/04/06|3|UK Salary
 Payment|Periodicity|Calendar Month
MERGE|BalanceAdjustmentValue|UK LDG|Salary Adjustment|UK Engineering Monthly||2022/04/06|4|UK Salary
 Payment|Periodicity Conversion Rule|Standard Rate Annualized
```

Guidelines for Adjusting National Insurance Balances for the UK

This balance adjustment example creates a batch header and group for an employee to adjust National Insurance category balances.

The predefined element NI Employee Adjustment is available for adjusting balances where required.

You can review the element's input values and the balances it feeds in the **Elements** task. There are several input values for this element and each feed one or more balances. You need to create a balance adjustment line for each required input value. If you need to adjust the individual category balances because an incorrect NI category has been processed for the employee, a separate set of adjustment lines must be supplied for each category. The context in these cases will be the NI category, which you need to provide.

The following input values are required for the NI Employee Adjustment element:

- Category
- Pension
- Statutory Deduction Card Id
- NIable LEL
- NIable LEL Calculation
- NIable PT
- NIable UAP
- NIable UEL
- NIable AUEL

- NIable
- NIable for Off Payroll Worker
- Pay Value
- NI Employee UEL
- NI Employee AUEL
- NI Employee
- NI Employee UAP
- NI Employee for Off Payroll Worker
- Director

Example of Adjusting National Insurance Balances for the UK

In this example, the requirement is to reduce the balances for category D and increase the balance values for Category A.

Load the Balance Adjustment Header and Group

Use the BalanceAdjustmentHeader.dat file to create the batch header and group records.

```
METADATA|BalanceAdjustmentHeader|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|UK LDG|NI Adjustment

METADATA|BalanceAdjustmentGroup|LegislativeDataGroupName|BatchName|EffectiveDate|PayrollName|
ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|UK LDG|NI Adjustment|2022/04/06|UK Engineering Monthly||Y|Y
```

Load the Balance Adjustment Line and Values

The line record specifies the element used to adjust the balances.

Balance Adjustment Line Data

Line Sequence	1
Batch Name	NI Employee Adjustment
Payroll Name	UK Engineering Monthly
Consolidation Set Name	N/A
Effective Date	2022/04/06
Assignment Number	E8184963
Element Name	NI Employee Adjustment
Tax Reporting Unit Name	UK Engineering Ltd

Balance Adjustment Values Data

Adjustment Values for Category A:

Category	A
Pension	Not contracted out
Statutory Deduction Card id	300012010991367
Nlable LEL	150.00
Nlable LEL Calculation	120.00
Nlable PT	100.00
Nlable UAP	0.00
Nlable UEL	25.00
Nlable UEL	25.00
Nlable AUEL	0.00
Nlable	425.00
Nlable for Off Payroll Worker	0.00
Pay Value	450.00
NI Employee UEL	0.00
NI Employee AUEL	0.00
NI Employee	345.00
NI Employee UAP	0.00
NI Employee for Off Payroll Worker	0.00
Director	No

Adjustment Values for Category D:

Category	D
Pension	Contracted out
Statutory Deduction Card id	300012010991367
Nlable LEL	-150.00
Nlable LEL Calculation	-120.00
Nlable PT	-100.00
Nlable UAP	0.00
Nlable UEL	-25.00

NIable AUEL	0.00
NIable	-425.00
NIable for Off Payroll Worker	0.00
Pay Value	-450.00
NI Employee UEL	0.00
NI Employee AUEL	0.00
NI Employee	-345.00
NI Employee UAP	0.00
NI Employee for Off Payroll Worker	0.00
Director	No

Use the BalanceAdjusmentLine.dat file to create batch line and value records:

```
METADATA|BalanceAdjustmentLine|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|ElementName|AssignmentNumber|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|E8184963|UK Engineering Ltd

METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|ElementName|InputValueName|EntryValue
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|Category|A
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1| NI Employee
 Adjustment|Pension|Not Contracted Out
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1| NI Employee
 Adjustment|Statutory Deduction Card Id|300012010991367
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NIable LEL|150
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1| NI Employee
 Adjustment|NIable LEL Calculation|120
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NIable PT|100
Monthly||2022/04/06|1|NI Employee Adjustment|NIable UAP|0
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NIable UEL|25
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NIable|425
Monthly||2022/04/06|1|NI Employee Adjustment|NIable for Off Payroll Worker|0
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|Pay Value|450
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NI Employee UEL|0
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NI Employee AUEL|0
Monthly||2022/04/06|1|NI Employee Adjustment|NI Employee|345
M Monthly||2022/04/06|1|NI Employee Adjustment|NI Employee UAP|0
onthly||2022/04/06|1|NI Employee Adjustment|NI Employee for Off Payroll Worker |0

MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|Director|No

METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|ElementName|InputValueName|EntryValue
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|Category|D
```

```
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1| NI Employee
 Adjustment|Pension|Contracted Out
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1| NI Employee
 Adjustment|Statutory Deduction Card Id|300012010991367
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NIable LEL|-150
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1| NI Employee
 Adjustment|NIable LEL Calculation|-120
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NIable PT|-100
Monthly||2022/04/06|1|NI Employee Adjustment|NIable UAP|0
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NIable UEL|-25
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NIable|425
Monthly||2022/04/06|1|NI Employee Adjustment|NIable for Off Payroll Worker|0
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|Pay Value|-450
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NI Employee UEL|0
MERGE|BalanceAdjustmentValue|UK LDG|NI Adjustment|UK Engineering Monthly||2022/04/06|1|NI Employee
 Adjustment|NI Employee AUEL|0
Monthly||2022/04/06|1|NI Employee Adjustment|NI Employee|-345
M Monthly||2022/04/06|1|NI Employee Adjustment|NI Employee UAP|0
onthly||2022/04/06|1|NI Employee Adjustment|NI Employee for Off Payroll Worker |0
```

Benefits and Pensions

Overview of Benefits and Pensions for the UK

The Benefits and Pensions calculation card stores information about the benefits and pension schemes an employee is enrolled into.

For the UK, the Benefits and Pensions card is an integral part of the Pensions Automatic Enrolment solution, as it records the qualifying pension scheme an employee has been enrolled in as the result of the Pension Automatic Enrolment process.

There are normally two ways a Benefits and Pensions card can be attached to an employee's record:

- By the Pensions Automatic Enrolment process: When an employee is found to be eligible for Pensions Automatic Enrolment by the Pensions Automatic Enrolment process, a Benefits and Pensions card is generated automatically, based on the default qualifying scheme.
- Created manually in the user interface by the Payroll Manager or Payroll Administrator.

There are cases where the Benefits and Pensions card may need to be created or updated in bulk:

During Data Migration:

- In order to process pension deductions accurately for those employees who are already enrolled in a pension scheme (whether qualifying or not), the pension contribution information needs to be brought forward to the Oracle Payroll Cloud.

Ongoing Bulk Updates:

- Bulk upload of pension rates: When employees are enrolled into a pension scheme (whether qualifying or not), employees may be allowed to choose the rate at which they want to contribute (either using Oracle Benefits or

a third party). These rates are stored as an override on the employee Benefits and Pensions card and need to be updated to ensure the correct contribution is calculated (this is true for both employee and employer rate).

Considerations and Prerequisites

- One element must be created for each Pension scheme for which contributions need to be processed through the Oracle Payroll Cloud.
- Eligibility for those elements must also be created.

Benefits and Pensions Card Record Types

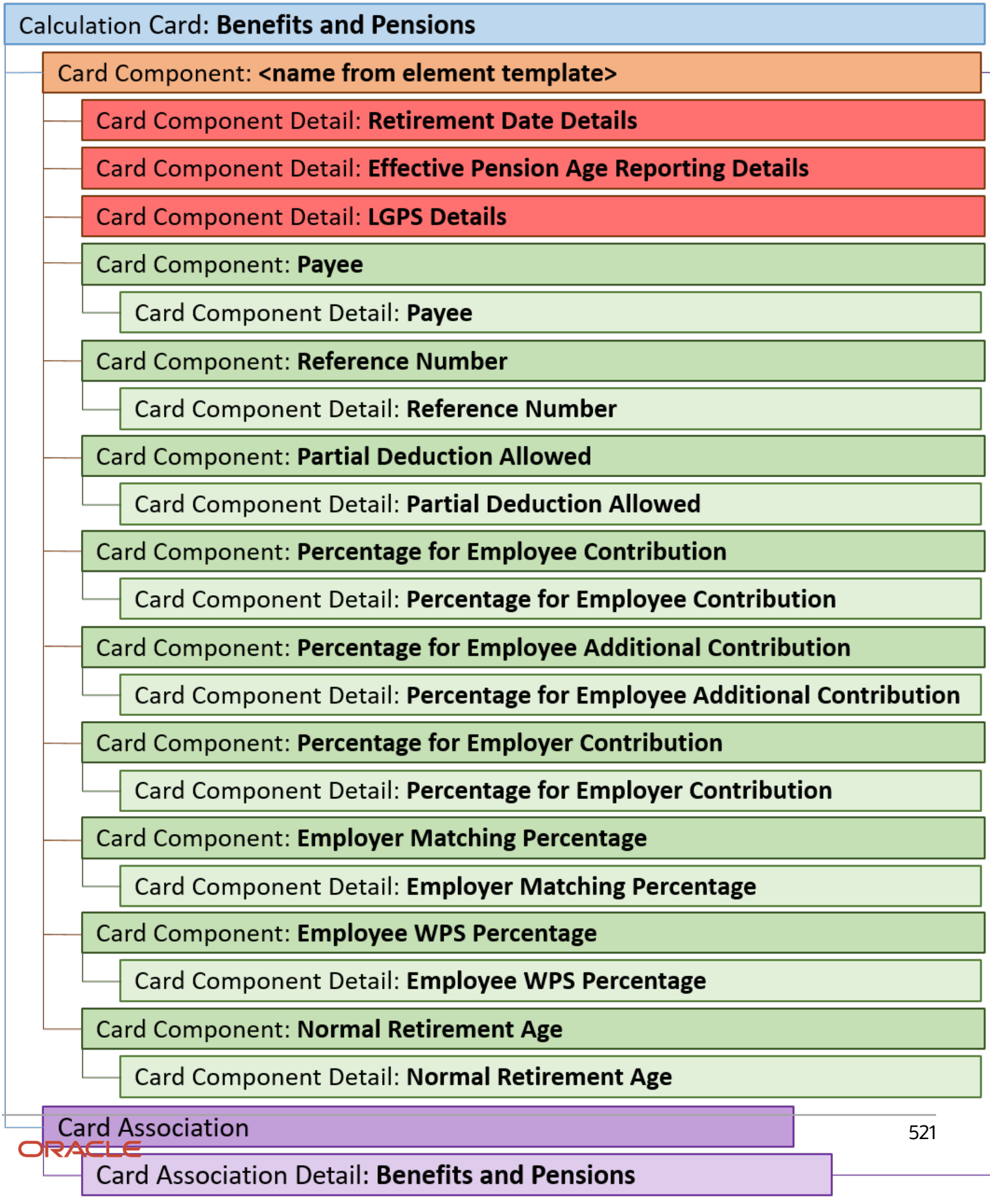
The Benefits and Pensions card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

UK Benefits and Pensions utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Calculation Value Definition	Allows the creation of value definitions so that overriding values can be specified on the card component.	CalculationValueDefinition
Enterable Calculation Value	Used to specify an overriding value for each calculation value definition.	EnterableCalculationValue
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Benefits and Pensions Calculation Card Hierarchy

The hierarchy of Calculation Card components applicable to Benefits and Pensions are described in this diagram:


Calculation Card: Benefits and Pensions
Card Component: <name from element template>
Card Component Detail: Retirement Date Details
Card Component Detail: Effective Pension Age Reporting Details
Card Component Detail: LGPS Details
Card Component: Payee
Card Component Detail: Payee
Card Component: Reference Number
Card Component Detail: Reference Number
Card Component: Partial Deduction Allowed
Card Component Detail: Partial Deduction Allowed
Card Component: Percentage for Employee Contribution
Card Component Detail: Percentage for Employee Contribution
Card Component: Percentage for Employee Additional Contribution
Card Component Detail: Percentage for Employee Additional Contribution
Card Component: Percentage for Employer Contribution
Card Component Detail: Percentage for Employer Contribution
Card Component: Employer Matching Percentage
Card Component Detail: Employer Matching Percentage
Card Component: Employee WPS Percentage
Card Component Detail: Employee WPS Percentage
Card Component: Normal Retirement Age
Card Component Detail: Normal Retirement Age
Card Association
Card Association Detail: Benefits and Pensions

The Benefits and Pension card component utilizes three flexfield contexts which are loaded using the Component Detail record type. There are a number of value definitions supported which are loaded using the Calculation Value Definition and Enterable Calculation Value record types.

Mapping Calculation card Components to the Responsive User Interface

The Pensions section of the Benefits and Pension calculation card shows:

- The name of the card component.
- All flexfield segments of the Component Details applicable to the pension type being viewed
- All value definitions applicable to the pension type being viewed.
- The card association assignments.

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Flexfield Data

Guidelines for Loading Benefits and Pensions for the UK

You must provide one Benefits and Pensions record for every UK employee you are capturing Pensions information for.

Even if you are updating an existing Benefits and Pensions card and the calculation card itself isn't being updated, still include the calculation card record to group other related data supplied in the file.

Benefits and Pensions Calculation Card Attributes

The Benefits and Pensions calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Benefits and Pensions calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Benefits and Pensions'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

These attributes are supplied against the CalculationCard discriminator. Always supply the CalculationCard record, even when updating an existing card.

Card Components Attributes

The Benefits and Pensions card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Benefits and Pensions card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Benefits and Pensions calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card component. This must be after or equal to the EffectiveStartDate on the Benefits and Pensions calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Provide the name of the component as generated by the element template.
ComponentSequence	N/A	Specify a sequence number to uniquely identify this card component.
Context1		The Pension Payroll ID used to uniquely identify employee's pension enrolment.

These attributes are supplied against the CardComponent discriminator.

Component Detail Attributes

The Benefits and Pensions card component uses three flexfield contexts. In addition to the common attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Retirement Date Details (ORA_HRX_GB_PS_RETIRE_DATE)
- Effective Pension Age Reporting Details (ORA_HRX_GB_PS_CSP_EPA_PENSION)
- LGPS Details (ORA_HRX_GB_LGPS_DETAILS)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Benefits and Pensions card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Benefits and Pensions card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'ORA_HRX_GB_PS_RETIRE_DATE'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
FLEX:Deduction DeveloperDF		Supply the same value as for the DirInformationCategory attribute.

These attributes are supplied against the ComponentDetail discriminator.

Value Definitions for Benefits and Pensions

The Benefits and Pensions card component uses these value definitions:

Value Definition Name	Functional Description	Type of Scheme
Payee	Third party organisation to which the pension contributions are paid.	All
Reference Number	Optional employee reference number with the payee.	All
Partial Deduction Allowed	Indicates whether partial deduction is allowed.	All
Percentage for Employee Contribution	Employee's contribution percentage	Private sector schemes and partnership schemes
Percentage for Employee Additional Contribution	Employee's additional contribution percentage	All additional contributions
Percentage for Employer Contribution	Employer's contribution percentage (overrides the value stored in the Calculation Value Definition if it exists)	Private sector schemes
Employer Matching Percentage	Used for partnership schemes only to indicate the percentage an employer may choose to contribute the match the employee's contribution rate.	Partnership schemes only
Employee WPS Percentage	Used for Alpha only to indicate the employee's WPS rate. It must be a multiple of 1.5	Alpha only
Normal Retirement Age	N/A	Private sector schemes

Calculation Value Definition Attributes

The Calculation Value Definition specifies the value definition name. Supply these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Benefits and Pensions card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent Benefits and Pensions card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent Benefits and Pensions card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

These attributes are supplied against the CalculationValueDefinition discriminator.

Enterable Calculation Value Attributes

The Enterable Calculation Value provides the override value for the value definition. Supply these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

These attributes are supplied against the EnterableCalculationValue discriminator.

Associating the Benefits and Pensions Calculation Card

The card association records associate the Benefits and Pension card component with the employee's tax reporting unit.

> **Note:** When creating a Benefits and Pensions card via Manage Calculation Card UI, the card association is non-enterable. However, when loading using HCM Data Loader, supply both the Card Association and Card Association Detail records.

Card Association Attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	AssignmentNumber, CardSequence, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Benefits and Pensions card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Benefits and Pensions calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Benefits and Pensions calculation card.

These attributes are supplied against the CardAssociation discriminator.

Card Association Detail Attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	AssignmentNumber, CardSequence, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
DirCardId(SourceSystemId)	AssignmentNumber, CardSequence, DirCardDefinitionName, LegislativeDataGroupName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Identify the Benefits and Pensions card component this association is for.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

These attributes are supplied against the CardAssociationDetail discriminator.

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Flexfield Data

Example of Creating a Benefits and Pensions Card for the UK

Refer to the Guidelines for Loading Benefits and Pensions of the United Kingdom topic for details of the attributes, flexfield context and value definition to supply.

This example create the Benefits and Pensions calculation card for employee assignment E453267 on 1st May 2019. The CalculationCard.dat file is used to upload Benefits and Pensions calculation cards using HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|BP_453267|UK LDG|Benefits and Pensions|2019/05/01|E453267

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|BP_453267_C1|UK LDG|BP_453267|2019/05/01|Pension_Element|6748311A

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName
```

```
MERGE|CalculationValueDefinition|VISION|BP_453267_C1_VD1|UK LDG|BP_453267_C1|2019/05/01|Pension_Element|
Payee

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|VISION|BP_453267_C1_VD1|UK LDG|BP_453267_C1_VD1|2019/05/01|Pension_Element|
Payee|Legal and General

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate
MERGE|CardAssociation|VISION|BP_453267_AS|UK LDG|BP_453267|2019/05/01

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|BP_453267_AS|BP_453267_C1|BP_453267_AS|2019/05/01|E453267
```

Example of Creating a Benefits and Pensions Card for Local Government Pension Scheme

This example creates a Benefits and Pensions card for a Local Government Pension Scheme (LGPS).

The CalculationCard.dat file is used to upload Benefits and Pensions calculation cards using HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|BP_483461|UK LDG|Benefits and Pensions|2019/01/01|E483461

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|BP_483461_LGPS|UK LDG|BP_483461|2019/01/01|LGPS Element|6187443A

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|FLEX:Deduction
 Developer DF|MemberOf5050Section(Deduction Developer DF=ORA_HRX_GB_LGPS_DETAILS)
MERGE|ComponentDetail|VISION|BP_483461_LGPS_PREL|UK LDG|BP_483461_LGPS|2019/01/01|LGPS Element|
ORA_HRX_GB_LGPS_DETAILS|ORA_HRX_GB_LGPS_DETAILS|Y

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate
MERGE|CardAssociation|VISION|BP_483461_AS|UK LDG|BP_483461|2019/01/01

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|BP_483461_ASD|BP_483461_LGPS|BP_483461_AS|2019/01/01|E483461
```

Court Orders and Student Loans

Overview of Court Orders and Student Loans Cards for the UK

The Court Orders and Student Loans card retains information for employees who are subject to court order or educational loan deductions (either Student or Postgraduate loans).

As such, this card type isn't generated automatically but must be created for those employees subject to these types of deductions.

The Court Orders and Student Loans card may need to be created in bulk:

During Data Migration:

- Employee's court orders and educational loans must be migrated to the Oracle Payroll Cloud, to ensure that appropriate deductions are made.

Ongoing Bulk Updates:

- New hires may have student loans that need to be uploaded in bulk.

It's advised to have a good knowledge of the Court Orders and Student Loans card in order to bulk-load data using HCM Data Loader. Refer to Oracle Fusion HCM (United Kingdom): Court Orders and Student Loans Implementation and Functional Considerations (MOS DOC ID: 2009287.1) for further information.

Considerations and Prerequisites

- Use the Manage Element task to create Court Orders and Student Loans elements. When you create the element, a card component of the same name as the element is created for use on the Court Orders and Student Loans card.
- There are several types of court order that can be recorded. One element must be created for each type of court order, student loan and postgraduate loan.
- Eligibility for the court order and student loan elements must be created.
- Create third-party payee and third-party payment methods for the people and organizations the payment of the court order is payable to.
- Court order issuing authorities must e added to the lookup type ORA_HRX_GB_CO_ISSUE_AUTHORITY.

The shape of the Court Orders and Student Loans card differs for court orders and student / postgraduate loans. Refer to the relevant topic for guidance on how to load these cards.

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Flexfield Data

Guidelines for Creating Court Orders for the UK

Refer to the Overview of Court Orders and Student Loans for the United Kingdom topic for considerations and prerequisites for bulk-loading Court Orders and Student Loans calculation cards.

Court Order Record Types

The Court Orders and Student Loans card is created with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

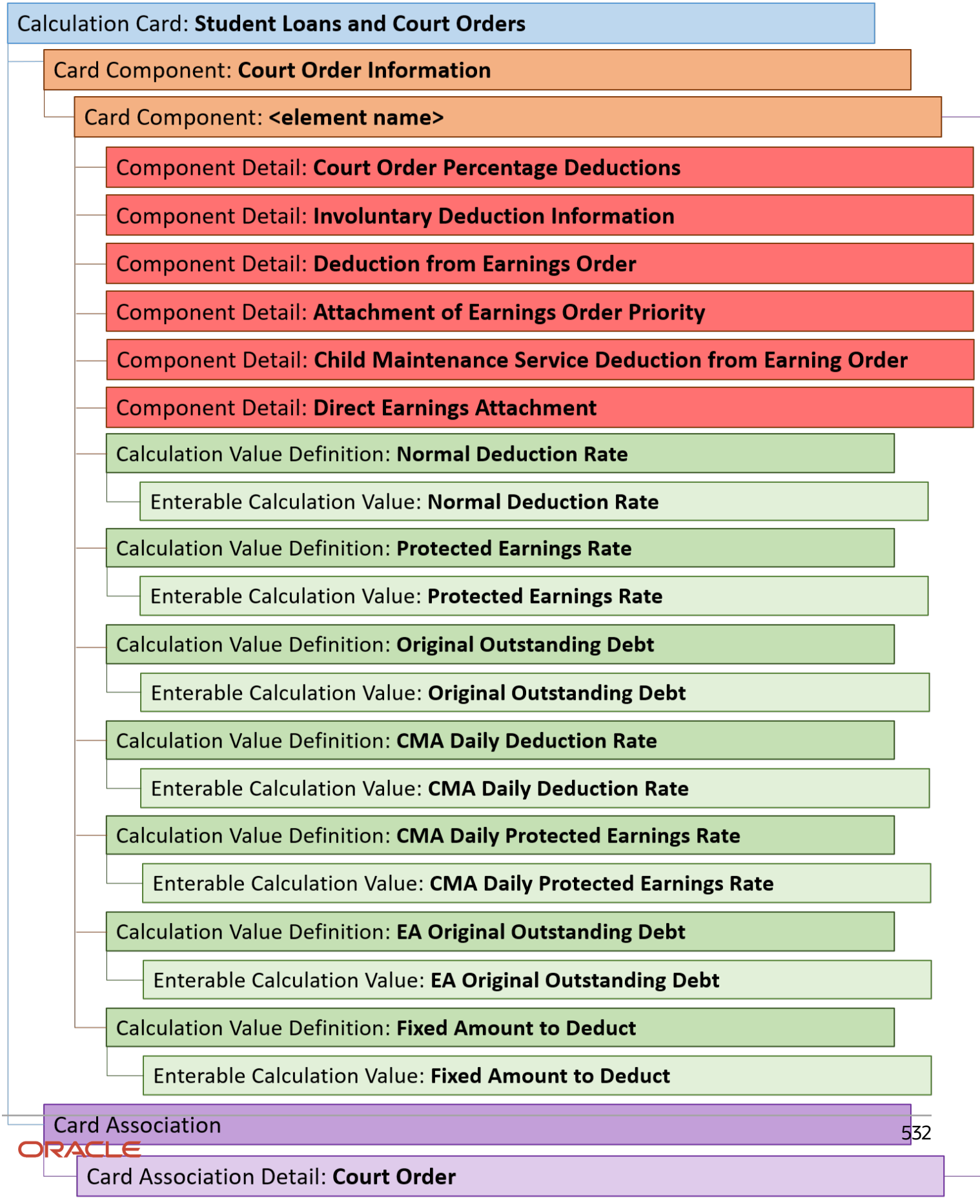
These record types are used when creating a calculation card to record court order information.

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card.	CardComponent
Component Detail	Used to capture data in the flexfield segments supported by each card component.	ComponentDetail
Calculation Value Definition	Specifies the value definitions that will be overridden on the calculation card.	CalculationValueDefinition
Enterable Calculation Value	Defines the overriding value for each calculation value definition.	EnterableCalculationValue
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Court Orders Card Hierarchy

The card shape for a court order differs not only from student and postgraduate loans, but also depending upon the type of court order.

This diagram describes all records that are relevant to all court order types:


Calculation Card: Student Loans and Court Orders
Card Component: Court Order Information
Card Component: <element name>
Component Detail: Court Order Percentage Deductions
Component Detail: Involuntary Deduction Information
Component Detail: Deduction from Earnings Order
Component Detail: Attachment of Earnings Order Priority
Component Detail: Child Maintenance Service Deduction from Earning Order
Component Detail: Direct Earnings Attachment
Calculation Value Definition: Normal Deduction Rate
Enterable Calculation Value: Normal Deduction Rate
Calculation Value Definition: Protected Earnings Rate
Enterable Calculation Value: Protected Earnings Rate
Calculation Value Definition: Original Outstanding Debt
Enterable Calculation Value: Original Outstanding Debt
Calculation Value Definition: CMA Daily Deduction Rate
Enterable Calculation Value: CMA Daily Deduction Rate
Calculation Value Definition: CMA Daily Protected Earnings Rate
Enterable Calculation Value: CMA Daily Protected Earnings Rate
Calculation Value Definition: EA Original Outstanding Debt
Enterable Calculation Value: EA Original Outstanding Debt
Calculation Value Definition: Fixed Amount to Deduct
Enterable Calculation Value: Fixed Amount to Deduct
Card Association
Card Association Detail: Court Order

Two card components are always required regardless of the court order type. Court Order Information is a parent card component to the second card component which is named after the element created for the court order type. The Court Orders component card name is defined by the element template.

Supply a Component Detail record for each flexfield context applicable to the court order type. Supply a Calculation Value Definition and Enterable Calculation Value record for each override value applicable to the court order type.

The Student Loans and Court Orders calculation card is associated with the employee's tax reporting unit by the card association and card association detail record types.

The following describes the attributes to supply for each record type.

Calculation Card Attributes

The Court Order and Student Loans calculation card record uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Court Orders and Student Loans calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Court Orders and Student Loans'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

These attributes are supplied against the CalculationCard discriminator. Always supply the CalculationCard record, even when updating an existing card.

Card Component Attributes

Supply two card component records, one for the Court Order Information card component, the other is named by the element created for the court order type.

Calculation Card: Student Loans and Court Orders

Card Component: Court Order Information

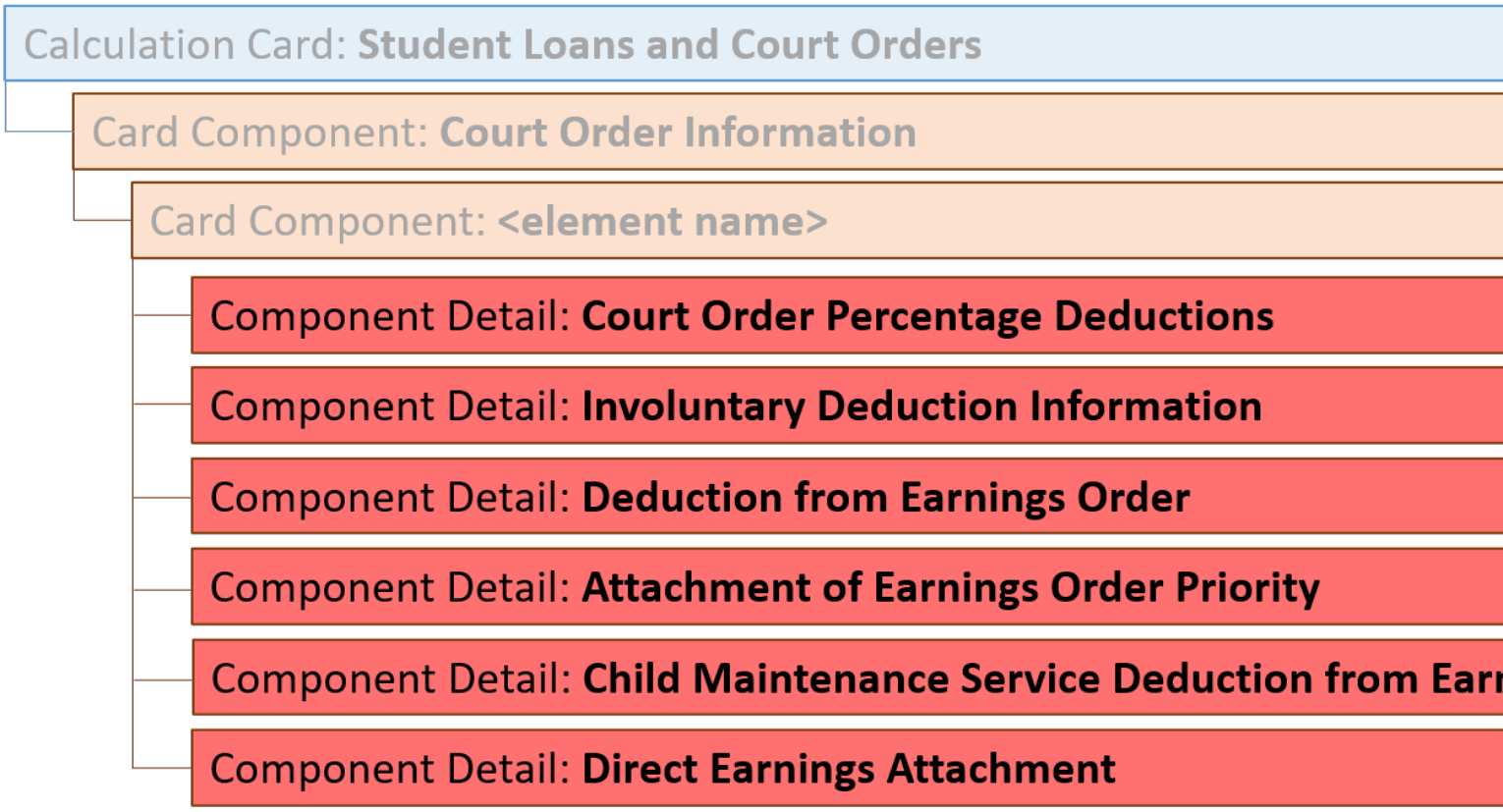
Card Component: <element name>

The Court Order and Student Loans card components use these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Court Orders and Student Loans card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders and Student Loans calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card component. This must be after or equal to the EffectiveStartDate on the Court Orders and Student Loans calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Supply 'Court Order Information' for one of the card components, and provide the name of the element created for the court order type for the other card component
ParentDirCardCompId (SourceSystemId)	AssignmentNumber, CardSequence, DirCardDefinitionName, LegislativeDataGroupName, ParentComponentSequence, ParentDirCardCompDefNam	Don't supply a value for the Court Order Information record. For the card component named for the court order element, supply a reference to the Court Order Information card component.
ComponentSequence	N/A	Specify a sequence number to uniquely identify this card component. Not required when supplying source keys.
Context1	N/A	The court order reference, as provided on the official notification.

These attributes are supplied against the CardComponent discriminator.

Component Detail Flexfield Contexts Applicable by Court Order Type



The type of court order determines the flexfield context to supply:

Attachment of Earning Order (AEO) Fines

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Court Order Percentage Deductions (ORA_HRX_GB_PCT)

Attachment of Earning Order (AEONI) Fines Northern Ireland

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Attachment of Earning Order (ORA_HRX_GB_AEO)

Attachment of Earning Order (AEO) Non-Priority

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Attachment of Earning Order (ORA_HRX_GB_AEO)

Attachment of Earning Order (AEO) Non-Priority Northern Ireland

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Attachment of Earning Order (ORA_HRX_GB_AEO)

Attachment of Earning Order (AEO) Priority

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Attachment of Earning Order (ORA_HRX_GB_AEO)

Attachment of Earning Order (AEO) Priority Northern Ireland

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Attachment of Earning Order (ORA_HRX_GB_AEO)

Child Maintenance Service Deduction from Earnings Order (CMSDEO)

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Child Maintenance Service Deduction from Earning Order (ORA_HRX_GB_CMSDEO)

Conjoined Arrestment Order (CAO)

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Court Order Percentage Deductions (ORA_HRX_GB_PCT)

Council Tax Attachment of Earnings Order (CTAEO)

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Court Order Percentage Deductions (ORA_HRX_GB_PCT)

Current Maintenance Arrestment (CMA)

- Involuntary Deduction Information (ORA_HRX_GB_INVD)

Deduction from Earnings Order (DEO)

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Deduction from Earnings Order (ORA_HRX_GB_DEO)

Direct Earnings Attachment (DEA)

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Attachment of Earning Order (ORA_HRX_GB_AEO)
- Direct Earnings Attachment (ORA_HRX_GB_DEA)

Earnings Arrestment (EA)

- Involuntary Deduction Information (ORA_HRX_GB_INVD)
- Court Order Percentage Deductions (ORA_HRX_GB_PCT)

Use the Component Detail record type to supply flexfield data.

Component Detail Attributes

The common attributes for all Component Detail record are as follows:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent card component should be referenced using the same key type used to identify the parent record. This is the card component that is named by the element created for the court order type. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parentcard component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'ORA_HRX_GB_INVD'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

When creating component detail records, refer to the card component that is named after the element created for the court order type as the parent card component. These attributes are supplied against the ComponentDetail discriminator.

Note: You can find the flexfield segment attribute names for each flexfield context using the View Business Objects task.

Value Definitions Applicable by Court Order Type

The type of court order determines the value definitions to supply:

Attachment of Earning Order (AEO) Fines

- Original Outstanding Debt

Attachment of Earning Order (AEONI) Fines Northern Ireland

- Original Outstanding Debt

Attachment of Earning Order (AEO) Non-Priority

- Nominal Deduction Rate
- Original Outstanding Debt

Attachment of Earning Order (AEO) Non-Priority Northern Ireland

- Nominal Deduction Rate
- Original Outstanding Debt

Attachment of Earning Order (AEO) Priority

- Nominal Deduction Rate
- Protected Earnings Rate
- Original Outstanding Debt

Attachment of Earning Order (AEO) Priority Northern Ireland

- Nominal Deduction Rate
- Protected Earnings Rate
- Original Outstanding Debt

Child Maintenance Service Deduction from Earnings Order (CMSDEO)

- Nominal Deduction Rate (all frequencies)

Conjoined Arrestment Order (CAO)

- CMA Daily Deduction Rate
- CMA Daily Protected Earnings Rate

Council Tax Attachment of Earnings Order (CTAEO)

- Original Outstanding Debt

Council Tax Attachment of Earnings Order Wales (CTAEOC)

- Original Outstanding Debt

Current Maintenance Arrestment (CMA)

- CMA Daily Deduction Rate

- CMA Daily Protected Earnings Rate

Deduction from Earnings Order (DEO)

- Nominal Deduction Rate
- Protected Earnings Rate

Direct Earnings Attachment (DEA)

- EA Original Outstanding Debt
- Fixed Amount to Deduct

Earnings Arrestment (EA)

- Original Outstanding Debt

Calculation Value Definition Attributes

The Calculation Value Definition specifies the value definition name. When creating calculation value definition records, refer to the card component that is named after the element created for the court order type as the parent card component.

Supply these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent card component should be referenced by using the same key type used to identify the card component. This is the card component that is named by the element created for the court order type. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent card component. Specify the name of the element created for the court order type.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this court order type are listed above.

These attributes are supplied against the CalculationValueDefinition discriminator.

Enterable Calculation Value Attributes

The Enterable Calculation Value provides the override value for the value definition. Supply these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes.
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

These attributes are supplied against the EnterableCalculationValue discriminator.

Associating the Court Orders and Student Loans Calculation Card

The card association record associates the Court Orders and Student Loans card component with a tax reporting unit. The card association details record allows Court Orders components to be associated with an employee assignment.

Card Association Attributes

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	AssignmentNumber, CardSequence, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Court Orders and Student Loans card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders and Student Loans calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Tax Withholding calculation card.
TaxReportingUnitName	N/A	The name of the employee's tax reporting unit to associate with the Court Orders and Student Loans calculation card. If source keys are used to identify the card association, you must also specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

These attributes are supplied against the CardAssociation discriminator.

Card Association Detail Attributes

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	AssignmentNumber, CardSequence, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
DirRepCardId(SourceSystemId)	AssignmentNumber, CardSequence, DirCardDefinitionName, LegislativeDataGroupName	The parent card association record should be identified by using the same key type supplied to against the card association.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Identify the card component this association is for. This is the card component that is named by the element created for the court order type.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

These attributes are supplied against the CardAssociationDetail discriminator.

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Flexfield Data

Example of Creating a Court Order card for the UK

Refer to the Guidelines for Creating Court Orders for the United Kingdom topic for details of the attributes, flexfield contexts and value definitions to supply for court orders.

This CalculationCard.dat file creates a new Court Orders and Student Loans card to record an Earning Arrestment (EA) court order for employee assignment E451388, starting on Jan 1st 2022.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|COSL_451388|UK LDG|Court Orders and Student Loans|2022/01/01|E451388

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|ParentDirCardCompId(SourceSystemId)|Context1
MERGE|CardComponent|VISION|CO_451388|UK LDG|COSL_451388|2022/01/01|Court Order Information||
MERGE|CardComponent|VISION|EA_451388|UK LDG|COSL_451388|2022/01/01|Earning Arrestment|CO_451388|15UKCORRF

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer DF|
DirInformationCategory|_DATE_OF_ISSUE(Deduction Developer DF=ORA_HRX_GB_INVD)|_DATE_OF_RECEIPT(Deduction
 Developer DF=ORA_HRX_GB_INVD)|_ISSUING_AUTHORITY_Display(Deduction Developer DF=ORA_HRX_GB_INVD)|
_THIRD_PARTY_PAYEE_Display(Deduction Developer DF=ORA_HRX_GB_INVD)|_REPORTING_REFERENCE(Deduction Developer
 DF=ORA_HRX_GB_INVD)|_PRIMARY_PAYROLL_Display(Deduction Developer DF=ORA_HRX_GB_PCT)
MERGE|ComponentDetail|VISION|EA_451388_INV|UK LDG|EA_451388|2022/01/01|Earning Arrestment|ORA_HRX_GB_INVD|
ORA_HRX_GB_INVD|2016/01/01|2016/01/05|Reading|UK REG CHQ TPPMQA|REF01|
MERGE|ComponentDetail|VISION|EA_451388_PCT|UK LDG|EA_451388|2022/01/01|Earning Arrestment|ORA_HRX_GB_PCT|
ORA_HRX_GB_PCT||||||UK Monthly

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName
```

```
MERGE|CalculationValueDefinition|VISION|EA_451388_OOD|UK LDG|EA_451388|2022/01/01|Earning Arrestment|
Original Outstanding Debt

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|VISION|EA_451388_OOD|UK LDG|EA_451388_OOD|2022/01/01|Original Outstanding
 Debt|2000

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|COSL_451388_TRU|UK LDG|COSL_451388|2022/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|COSL_451388_ASD|EA_451388|COSL_451388_TRU|2022/01/01|E451388
```

Guidelines for Creating Student and Postgraduate Loan for the UK

Refer to the Overview of Court Orders and Student Loans for the United Kingdom topic for considerations and prerequisites for bulk-loading Court Orders and Student Loans calculation cards.

Student or Postgraduate Loans Record Types

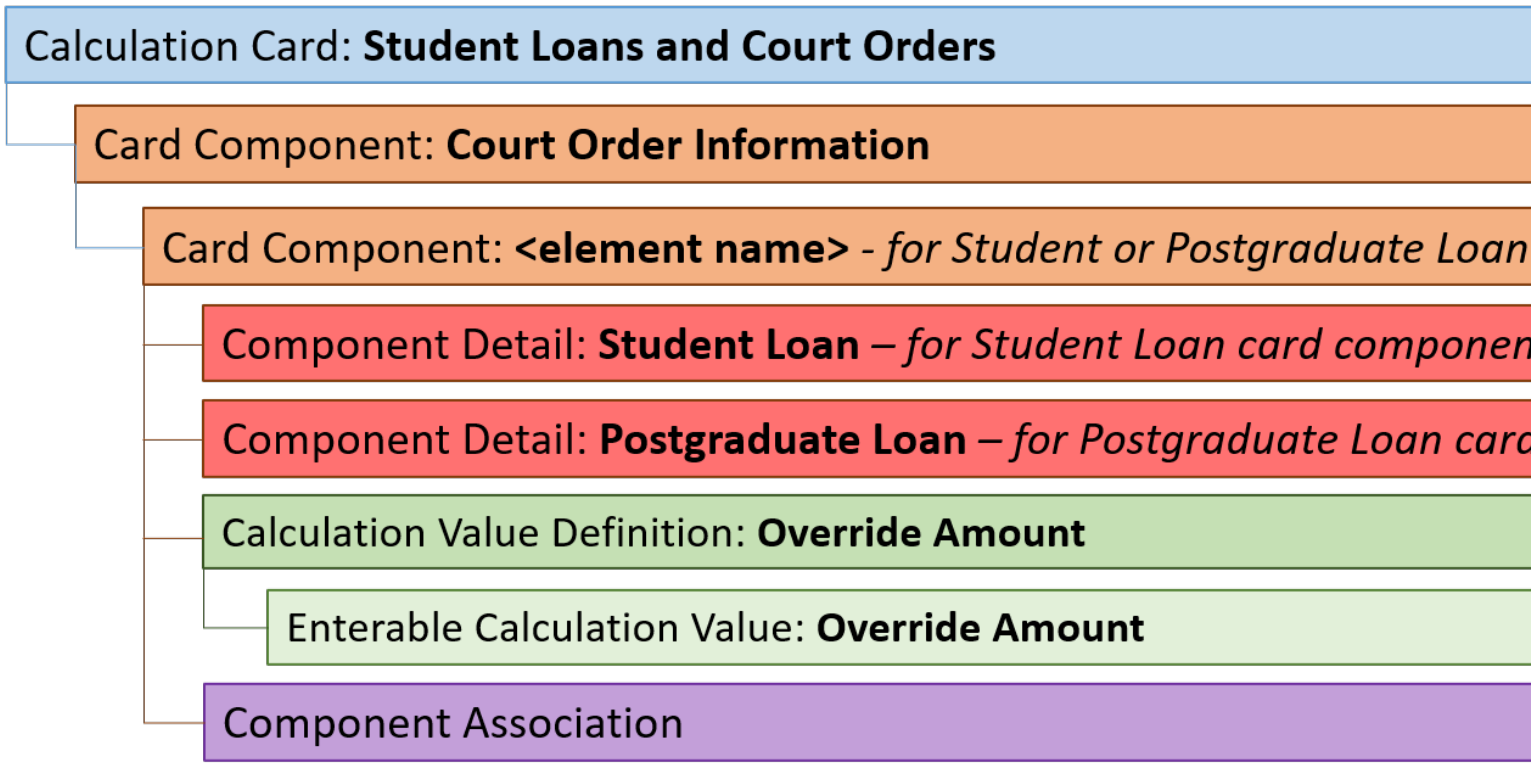
The Court Orders and Student Loans card is created with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

These record types are used when creating a calculation card to record educational loan information.

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card.	CardComponent
Component Detail	Used to capture data in the flexfield segments supported by each card component.	ComponentDetail
Calculation Value Definition	Specifies the value definitions that will be overridden on the calculation card.	CalculationValueDefinition
Enterable Calculation Value	Defines the overriding value for each calculation value definition.	EnterableCalculationValue
Component Association	Associates the educational loan card component with the Tax Reporting Unit the employee reports to.	ComponentAssociation

Student or Postgraduate Loans Card Hierarchy

The card shape for an educational loan differs from that of a court order. This diagram describes all records that are relevant to all court order types:



Two card components are always required. Court Order Information is a parent card component to the card component which is named after the element created for the student or postgraduate loan.

Supply a Component Detail record for the flexfield context applicable to the student or postgraduate loan.

Supply a Calculation Value Definition and Enterable Calculation Value record for the Override Amount value definition.

The following describes the attributes to supply for each record type.

Calculation Card Attributes

The Court Order and Student Loans calculation card record uses these attributes:

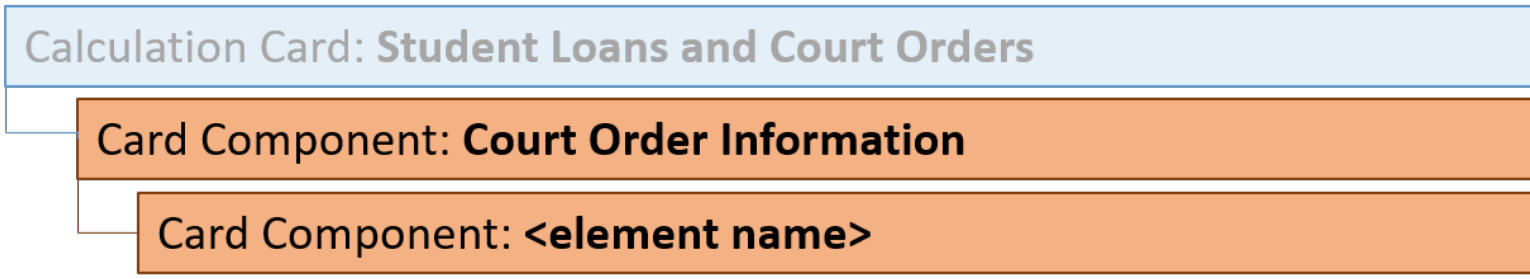
HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Court Orders and Student Loans calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Court Orders and Student Loans'.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

These attributes are supplied against the CalculationCard discriminator. Always supply the CalculationCard record, even when updating an existing card.

Card Component Attributes

Supply two card component records, one for the Court Order Information card component, the other is named by the element created for the educational loan type.



The Court Order and Student Loans card components use these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Court Orders and Student Loans card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders and Student Loans calculation card should be identified by using the same key type used to identify the calculation card. When using source keys,

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card component. This must be after or equal to the EffectiveStartDate on the Court Orders and Student Loans calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Supply 'Court Order Information' for one of the card components, and provide the name of the element created for the court order type for the other card component.
ComponentSequence	N/A	Specify a sequence number to uniquely identify this card component. Not required when supplying source keys.
Context1	N/A	The educational loan reference, as provided on the official notification.

These attributes are supplied against the CardComponent discriminator.

Component Detail Attributes

The flexfield context to supply is dependent on the type of educational loan:

Student Loan

- Student Loan (ORA_HRX_GB_SL)

Postgraduate Loan

- Postgraduate Loan (ORA_HRX_GB_PGL)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

The common attributes for all Component Detail record are as follows:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent card component should be referenced using the same key type used to identify the parent record. This is the card component that is named by the element created for the court order type. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parentcard component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'ORA_HRX_GB_INVD'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

These attributes are supplied against the ComponentDetail discriminator.

Note: You can find the flexfield segment attribute names for each flexfield context using the View Business Objects task.

Calculation Value Definition Attributes

The Calculation Value Definition specifies the value definition name.

When creating calculation value definition records, refer to the card component that is named after the element created for the court order type as the parent card component.

Supply these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName,	A unique identifier for the calculation value definition record. For new records supply the

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, DirCardCompDefName	source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent card component should be referenced by using the same key type used to identify the card component. This is the card component that is named by the element created for the court order type. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent card component. Specify the name of the element created for the court order type.
ValueDefinitionName	N/A	The name of the value being overridden. Specify 'Override Amount'.

These attributes are supplied against the CalculationValueDefinition discriminator.

Enterable Calculation Value Attributes

The Enterable Calculation Value provides the override value for the value definition. Supply these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes.
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

These attributes are supplied against the EnterableCalculationValue discriminator.

Associating the Educational Loan Card Component

The component association record associates the student /postgraduate loan card component with the tax reporting units the employee reports to.

Component Association Attributes

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	AssignmentNumber, CardSequence, ComponentSequence, DirCardCompDefName, DirCardDefinitionName, LegislativeDataGroupName, TaxReportingUnitName	A unique identifier for the educational loan component association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardCompId(SourceSystemId)	AssignmentNumber, CardSequence, ComponentSequence, DirCardCompDefName, DirCardDefinitionName, LegislativeDataGroupName	The parent educational loan card component should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Tax Withholding calculation cardeducational loan card component.
TaxReportingUnitName	N/A	The name of the tax reporting unit the employee reports to.

These attributes are supplied against the ComponentAssociation discriminator.

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Flexfield Data

Example of Creating Student Loan for the UK

Refer to the Guidelines for Creating Educational Loans for the United Kingdom topic for details of the attributes, flexfield contexts and value definitions to supply.

This CalculationCard.dat file creates a new Court Orders and Student Loans card to record details of a Student Loan for employee assignment E182643, starting on Jan 1st 2018.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|COSL_182643|UK LDG|Court Orders and Student Loans|2018/01/01|E182643

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|COI_182643|UK LDG|COSL_182643|2018/01/01|Court Order Information
MERGE|CardComponent|VISION|SL_182643|UK LDG|COSL_182643|2018/01/01|UK Student Loan

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer DF|
DirInformationCategory|_PLAN_TYPE(Deduction Developer DF=ORA_HRX_GB_SL)|_START_DATE(Deduction Developer
 DF=ORA_HRX_GB_SL)|_START_DATE_NOTICE(Deduction Developer DF=ORA_HRX_GB_SL)|_STOP_DATE(Deduction Developer
 DF=ORA_HRX_GB_SL)|_STOP_DATE_NOTICE(Deduction Developer DF=ORA_HRX_GB_SL)|_TEACHER_REPAYMENT(Deduction
 Developer DF=ORA_HRX_GB_SL)|_LAST_UPDATE_PROCESS_SEQUENCE(Deduction Developer DF=ORA_HRX_GB_SL)
MERGE|ComponentDetail|VISION|SL_182643_GB_SL|UK LDG|SL_182643|2018/01/01|UK Student Loan|ORA_HRX_GB_SL|
ORA_HRX_GB_SL|1|2018/01/01|2018/01/01||||10000

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|SL_182643_OA|UK LDG|SL_182643|2018/01/01|UK Student Loan|Override
 Amount

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|Value1
MERGE|EnterableCalculationValue|VISION|SL_182643_OA|UK LDG|SL_182643_OA|2018/01/01|1250

METADATA|ComponentAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|ComponentAssociation|VISION|SL_182643_ASOC|UK LDG|SL_182643|2018/01/01|UK TRU
```

Example of Creating Postgraduate Loan for the UK

Refer to the Guidelines for Creating Educational Loans for the United Kingdom topic for details of the attributes, flexfield contexts and value definitions to supply. This

CalculationCard.dat file creates a new Court Orders and Student Loans card to record details of a Postgraduate Loan for employee assignment E366231, starting on Jan 1st 2018.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
```

```
MERGE|CalculationCard|VISION|COSL_366231|UK LDG|Court Orders and Student Loans|2018/01/01|E366231

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|COI_366231|UK LDG|COSL_366231|2018/01/01|Court Order Information
MERGE|CardComponent|VISION|PGL_366231|UK LDG|COSL_366231|2018/01/01|Postgraduate Loan

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer DF|
DirInformationCategory|_START_DATE(Deduction Developer DF=ORA_HRX_GB_PGL)|_START_DATE_NOTICE(Deduction
 Developer DF=ORA_HRX_GB_PGL)|_STOP_DATE(Deduction Developer DF=ORA_HRX_GB_PGL)|_STOP_DATE_NOTICE(Deduction
 Developer DF=ORA_HRX_GB_PGL)|_LAST_UPDATE_PROCESS_SEQUENCE(Deduction Developer DF=ORA_HRX_GB_PGL)
MERGE|ComponentDetail|VISION|PGL_366231_GB_PBL|UK LDG|PGL_366231|2018/01/01|Postgraduate Loan|
ORA_HRX_GB_PGL|ORA_HRX_GB_PGL|2018/01/01|2018/01/01|||10000

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|PGL_366231_OA|UK LDG|PGL_366231|2018/01/01|Postgraduate Loan|
Override Amount

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|Value1
MERGE|EnterableCalculationValue|VISION|PGL_366231_OA|UK LDG|PGL_366231_OA|2018/01/01|1250

METADATA|ComponentAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|ComponentAssociation|VISION|PGL_366231_ASOC|UK LDG|PGL_366231|2018/01/01|UK TRU
```

Pension Automatic Enrolment

Overview of Pensions Automatic Enrolment Cards for the UK

Pension Automatic Enrolment cards store the employee's situation related to Pension's Automatic Enrolment process, such as the employee's eligibility for pension, which qualifying pension scheme they are enrolled into, and opt-in and opt-out details.

A default card is generated automatically when an employee record is created by the New Hire flow and the license is set to Payroll, or Payroll Interface.

Considerations and Prerequisites

By default, an employee's classification is set to Not Yet Assessed. If a Tax Reporting Unit (TRU) is provided for the employee during the New Hire flow, the Pensions Automatic Enrolment card is associated automatically to that TRU. The Pensions Automatic Enrolment process can then be run on a regular basis to assess the employee's eligibility to Pensions Automatic Enrolment and make all the relevant updates to the card. If the employee is found eligible, a Benefits and Pensions card is also created automatically by the process for the default Qualifying Pension Scheme.

There may be instances where mass upload is required for the Pensions Automatic Enrolment card:

During data migration:

- Employee's Pensions Automatic Enrolment must be uploaded to ensure that all relevant information about Pensions Automatic Enrolment is migrated from your legacy system. When you use HCM Data Loader to migrate the employee records a card is generated with the default Not Yet Assessed classification. This needs

to be updated to reflect the employee's current state, or they will be re-enrolled the first time the Pensions Automatic Enrolment process is executed in the Oracle Payroll Cloud.

Ongoing bulk updates:

- **Bulk Load of New Hire data:**

  If you use HCM Data Loader or interface with Taleo to bulk upload new hire information a default Pensions Automatic Enrolment card may be generated automatically. You should update auto-generated cards if the defaulted data isn't accurate or relevant.

- **Bulk Load of Pensions Automatic Enrolment Information:**

  If a third-party manages Pension Automatic Enrolment data, such as opt-in, opt-out and notification letters, information obtained from the third party must be uploaded into the Oracle Payroll cloud.

It is recommended that you have a good understanding of the Pensions Automatic Enrolment card and the information it contains before attempting to bulk load data for it. For further information see Oracle Fusion HCM (United Kingdom): Pensions Automatic Enrolment and Functional Considerations (MOS DOC ID: 2006584.1).

Prerequisites:

- You must have created the element eligibility for the Pensions Automatic Enrolment element.
- You must have created the Pension Deduction elements.
- When uploading Qualifying Pension Scheme via HCM Data Loader, you will need to ensure that the corresponding component exists in the Benefits and Pensions card.

Pensions Automatic Enrolment Card Record Types

The Pensions Automatic Enrolment card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

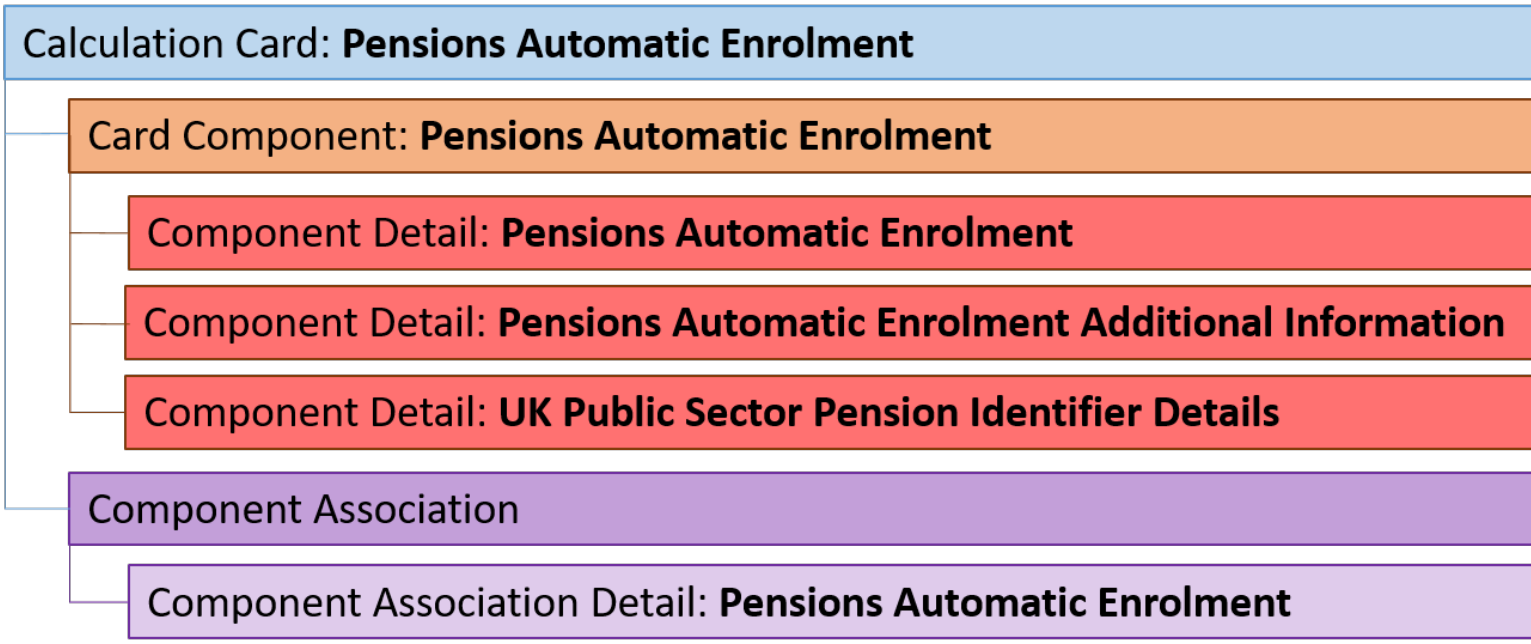
The UK Pensions Automatic Enrolment utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Component Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	ComponentAssociation

Component	Functional Description	File Discriminator
	Note: If there is only one TRU in your organization, the component association isn't mandatory: by default it will automatically be associated to the unique TRU. If you add another component to the card, it must be associated to a TRU.	
Component Association Detail	Associates card components with the employee's assignments. Note: When there are multiple components on the same card, each component must be associated to a TRU, and to Terms or Assignments.	ComponentAssociationDetails

Pensions Automatic Enrolment Calculation Card Hierarchy

The hierarchy of calculation card components applicable Pensions Automatic Enrolment are described in this diagram:



The Pensions Automatic Enrolment component card utilizes three flexfield contexts which are loaded using the Component Detail record type. The component is associated with the employee's tax reporting unit using the Component Association and Component Association Detail records types.

Mapping Calculation Card Components to the Responsive User Interface

The Pensions Automatic Enrolment Details are maintained using the card component detail records.

The TRU information is updated using the component association record. The Assignments associated with the card are defined using the Component Association Detail record.

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Pensions Automatic Enrolments for the UK
- Example of Loading Pensions Automatic Enrolments for the UK
- Guidelines for Loading Flexfield Data

Guidelines for Loading Pensions Automatic Enrolments for the UK

You must provide one Calculation Card record for every UK employee you are maintaining Pensions Automatic Enrolment data for.

Even if you are updating an existing Pensions Automatic Enrolment calculation card and calculation card itself isn't being updated, still include the calculation card record to group other related data supplied in the file.

Calculation Card Attributes

The Pensions Automatic Enrolment calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Pensions Automatic Enrolment calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Pensions Automatic Enrolment'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Calculation Component Attributes

You must provide a Pensions Automatic Enrolment card component for each tax reporting unit the employee reports to.

The Pensions Automatic Enrolment card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Pensions Automatic Enrolment card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Pensions Automatic Enrolment calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Pensions Automatic Enrolment card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Statutory Deductions calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName		The component definition name. Specify 'Pensions Automatic Enrolment'.
ComponentSequence	N/A	Specify '1'.

Component Detail Attributes

The Pensions Automatic Enrolment component card uses three flexfield contexts which are loaded using the Component Detail record type. Supply these, if relevant, along with the parent card component and calculation card records.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Pensions Automatic Enrolment (HRX_GB_PAE)
- Pensions Automatic Enrolment Additional Information (ORA_HRX_GB_PAE_ADNL)
- UK Public Section Pension Identifier Details (ORA_HRX_GB_PS_PENSION)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Pensions Automatic Enrolment card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Pensions Automatic Enrolment card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_GB_PAE'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Note: When referencing the overriding and active qualifying schemes, refer to the Cloud Customer Connect topic BI Publisher Report (HDL): UK Pension Qualifying Schemes topic for a report that extracts the qualifying scheme names and IDs by legislative data group. When supplying this information to the flexfield segments, you can supply the ID to the base attributes:

- _ACTIVE_QUALIFYING_SCHEME(Deduction Developer DF=HRX_GB_PAE)
- _OVERRIDING_QUALIFYING_SCHEME(Deduction Developer DF=HRX_GB_PAE)

Alternatively, supply the name to the translation attributes. However, this must be supplied in HDL files using base language of your environment:

- _ACTIVE_QUALIFYING_SCHEME_Display(Deduction Developer DF=HRX_GB_PAE) -
- _OVERRIDING_QUALIFYING_SCHEME_Display(Deduction Developer DF=HRX_GB_PAE)

Component Association Attributes

The Component Association record type associates the Pensions Automatic Enrolment card component with the employee's tax reporting unit.

The Component Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName, ComponentSequence	A unique identifier for the Pensions Automatic Enrolment card component association. For new card component associations supply the source key attributes. You can also identify card components associations with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Pensions Automatic Enrolment card component should be identified by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card component association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Pensions Automatic Enrolment calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Pensions Automatic Enrolment card

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		component. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Component Association Detail Attributes

The card association details record associates the Pensions Automatic Enrolment card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Component Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	DirCardDefinitionName, AssignmentNumber, LegislativeDataGroupName, CardSequence, TaxReportingUnitName, DirCardCompDefName, ComponentSequence, AssociationAssignmentNumber	A unique identifier for the component association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
DirRepCardId(SourceSystemId)	CardSequence, ComponentSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	The parent component association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Pensions Automatic Enrolment Cards for the UK
- Example of Loading Pensions Automatic Enrolments for the UK
- Guidelines for Loading Flexfield Data

Example of Loading Pensions Automatic Enrolments for the UK

This example create a Pensions Automatic Enrolment card for employee assignment E451388, starting on the 1st May 2019.

Source keys are used to identify each record in calculation card hierarchy.

Use the CalculationCard.dat file to upload the Pensions Automatic Enrolment card with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|PAE_451388|UK LDG|Pensions Automatic Enrolment|2019/05/01|E451388

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|PAE_451388_CC|UK LDG|PAE_451388|2019/05/01|Pensions Automatic Enrolment

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|FLEX:Deduction
 Developer DF|_EMPLOYEE_CLASSIFICATION(Deduction Developer DF=HRX_GB_PAE)|_ELIGIBLE_JOBHOLDER_DATE(Deduction
 Developer DF=HRX_GB_PAE)|_RE_ENROLMENT_ASSESSMENT_ONLY(Deduction Developer DF=HRX_GB_PAE)|
_REASON_FOR_EXCLUSION(Deduction Developer DF=HRX_GB_PAE)|_OVERRIDING_WORKER_POSTPONEMENT(Deduction
 Developer DF=HRX_GB_PAE)|_OVERRIDING_ELIGIBLEJH_POSTPMNT(Deduction Developer DF=HRX_GB_PAE)|
_OVERRIDING_QUALIFYING_SCHEME_Display(Deduction Developer DF=HRX_GB_PAE)|_ACTIVE_POSTPONEMENT_TYPE(Deduction
 Developer DF=HRX_GB_PAE)|_ACTIVE_POSTPONEMENT_RULE(Deduction Developer DF=HRX_GB_PAE)|
_ACTIVE_POSTPONEMENT_END_DATE(Deduction Developer DF=HRX_GB_PAE)|_ACTIVE_QUALIFYING_SCHEME_Display(Deduction
 Developer DF=HRX_GB_PAE)|_QUALIFYING_SCHEME_JOIN_METHOD(Deduction Developer DF=HRX_GB_PAE)|
_QUALIFYING_SCHEME_JOIN_DATE(Deduction Developer DF=HRX_GB_PAE)|_QUALIFYING_SCHEME_LEAVE_REASON(Deduction
 Developer DF=HRX_GB_PAE)|_QUALIFYING_SCHEME_LEAVE_DATE( Deduction Developer DF=HRX_GB_PAE)|
_OPT_OUT_PERIOD_END_DATE(Deduction Developer DF=HRX_GB_PAE)|_OPT_OUT_REFUND_DUE(Deduction
 Developer DF=HRX_GB_PAE)|_JOINING_SCHEME_PROCESSED_DATE(Deduction Developer DF=HRX_GB_PAE)|
_LEAVING_SCHEME_PROCESSED_DATE(Deduction Developer DF=HRX_GB_PAE)|_CLASSIFICATION_CHANGED_PROC_DT(Deduction
 Developer DF=HRX_GB_PAE)|_ACTIVE_POSTPONEMENT_PROC_DATE(Deduction Developer DF=HRX_GB_PAE)|
_PAYROLL_FOR_PAY_REF_PERIOD(Deduction Developer DF=HRX_GB_PAE)|_OVERRIDING_STAGING_DATE(Deduction
 Developer DF=HRX_GB_PAE)|_RE_ENROLMENT_DATE(Deduction Developer DF=HRX_GB_PAE)|
_QUALIFYING_SCHEME_START_DATE(Deduction Developer DF=HRX_GB_PAE)|_LETTER_TYPE_GENERATED(Deduction Developer
 DF=HRX_GB_PAE)|_LETTER_STATUS(Deduction Developer DF=HRX_GB_PAE)|_LETTER_TYPE_DATE(Deduction Developer
 DF=HRX_GB_PAE)|_SUBSEQUENT_COMMS_REQ(Deduction Developer DF=HRX_GB_PAE)|_LETTER_HISTORY (Deduction
 Developer DF=HRX_GB_PAE)
MERGE|ComponentDetail|VISION|PAE_451388_FLEX|UK LDG|PAE_451388_CC|2019/05/01|Pensions Automatic
 Enrolment|HRX_GB_PAE|HRX_GB_PAE|ELIGIBLEJH|2019/05/01|Y|||My_Premium_Element|||My_Premium_Element|
AUTOENROL|2019/05/01|||2019/06/30|Y|2019/05/01||2019/05/01|||||2019/05/01|ORA_HRX_GB_LETTER_E|
ORA_HRX_GB_LET_NOTYETGEN|||

METADATA|ComponentAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|ComponentAssociation|VISION|PAE_451388_AS|UK LDG|PAE_451388_CC|2019/05/01|UK Tax Reporting Unit

METADATA|ComponentAssociationDetail|SourceSystemOwner|SourceSystemId|DirRepCardId(SourceSystemId)|
DirCardCompId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|ComponentAssociationDetail|VISION|PAE_451388_ASD|PAE_451388_AS|PAE_451388_CC|2019/05/01|E451388
```

Statutory Deductions

Overview of Statutory Deductions for the UK

Statutory Deduction cards store all the information required to accurately compute Tax and National Insurance (NI) contributions for an employee, such as tax code, NI category and pension basis.

It also includes information required for the Real Time Information (RTI) reports. A default card is automatically generated when an employee record is created by the New Hire flow and the license is set to Payroll, or Payroll Interface.

One Statutory Deduction card must be created for each employee and the Tax Reporting Unit (TRU) that the employee reports to.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, Statutory Deduction cards are automatically created when a new employee is entered using the New Hire task with a set of default values specified at the PSU or the TRU level.

However, there may be cases where this information must be loaded in bulk:

During Data Migration:

Employee PAYE and NI information must be uploaded into Oracle Fusion Payroll, in order to ensure that contributions are calculated correctly. If HCM Data Loader is used to migrate employee records, a default statutory deduction card is automatically created. In most cases, the default won't reflect the employee's actual tax and NI information, and therefore the card must be updated.

A number of information fields must be provided for RTI purposes such as:

- Previous HMRC Payroll ID from your legacy system to inform HMRC of the change in Payroll ID.

  > **Note:** The new Oracle Cloud HRMC Payroll ID is automatically generated when a new Statutory Deduction card is created for the person. When the Full Payment Submission (FPS) is submitted for the first time from Oracle Cloud Payroll, it contains the mapping between the new and old HMRC Payroll ID. If the employment has already been filed with HMRC, it won't be reported as a new one.

- Information such as Late Hire Reason, Number of Hours Worked, New Hire Declaration, Pensioner Notification, or Expatriate Notification, if required.

Ongoing bulk updates:

Bulk loading of new hire data: If you have to do a mass upload of New Hire information, a default Statutory Deduction card and Pensions Automatic Enrolment card may be automatically generated (if the new hire records are created through HCM Data Loader or the interface with Taleo). In this case, you need to update the default card with the correct tax and NI information and create a component detail to store new hire information as requested for RTI reporting.

It is recommended to have a good understanding of the Statutory Deductions card and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS (United Kingdom): Payroll Implementation and Functional Considerations (Doc ID 1921464.1).

Note: Some validations that are enforced by the user interface can't be enforced whilst loading records via HCM Data Loader. We would therefore advise that you run either the Payroll Validation Report or the Diagnostic report after loading Statutory Deduction records via HCM Data Loader.

Statutory Deduction Card Record Types

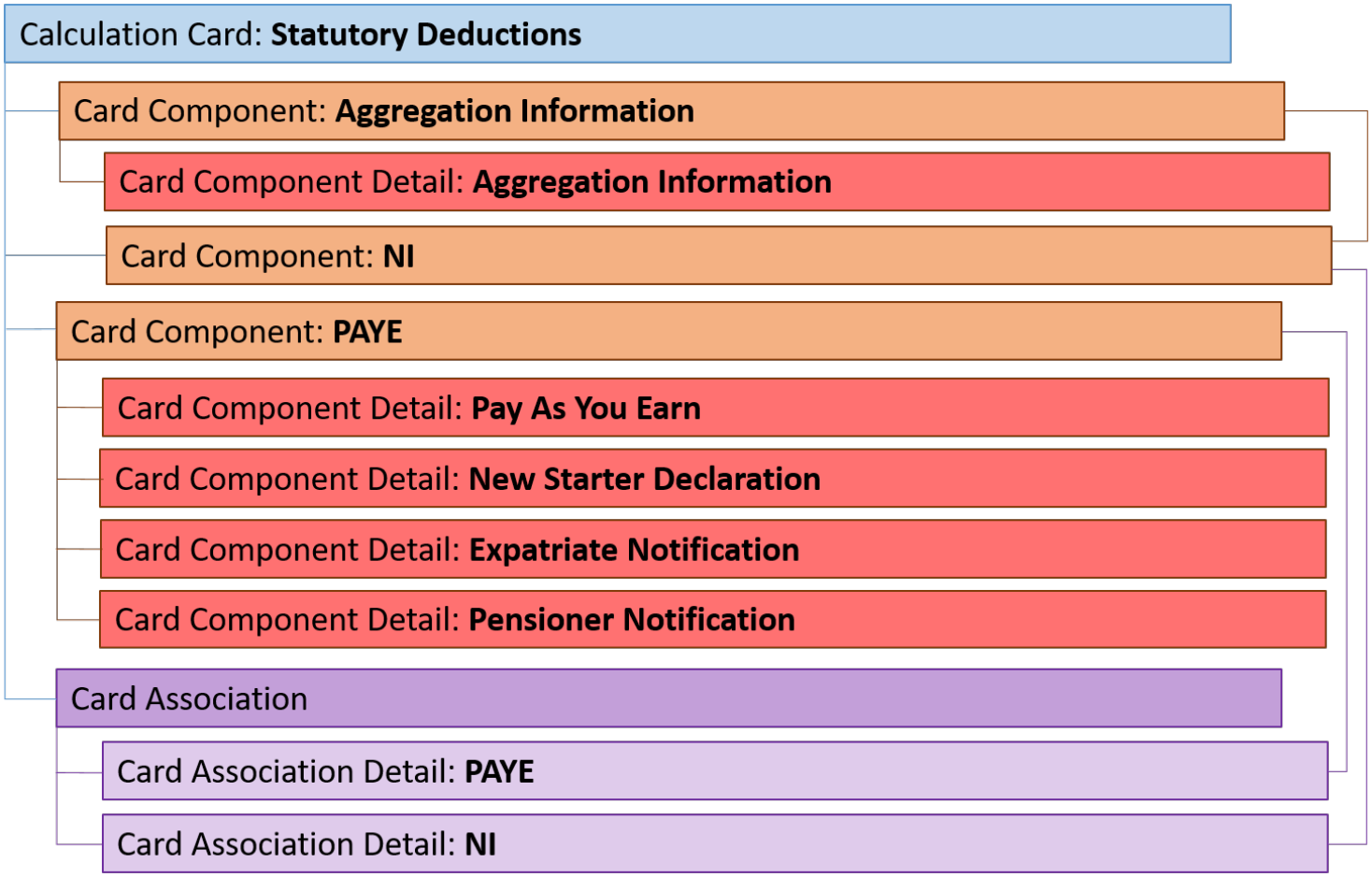
The Statutory Deduction card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The UK Statutory Deductions utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Statutory Deduction Calculation Card Hierarchy

The hierarchy of calculation card components applicable Statutory Deductions are described in this diagram:



A separate topic follows to describe how to provide data for the calculation card record and each card component.

Mapping Calculation card components to the responsive user interface

The TRU information is updated using the card association record.

Paye As You Earn and National Insurance Statutory Deductions summary information is updated using the PAYE and NI card components, with the Assignment information being defined on the card association detail records.

PAYE Details are updated using the Pay As You Earn card component detail record. On the PAYE Details page, you can also create one or more Notifications. The Notification Type enables you to select from one of the three component details: New Starter Declaration, Pensioner Notification or Expatriate Notification. These are updated using the appropriate Card Component Detail records under the PAYE card component.

NI Details are updated using the NI card component. NI Additional Info is updated using the Aggregation Information card component detail under the Aggregation Information card component. The Aggregation Information card component is not displayed on the user interface. However, it is part of the calculation card structure and needs to be supplied when creating a new Statutory Deductions card using HCM Data Loader. If multiple NI components exist, the Aggregation Information will be displayed against each NI component, though only one Aggregation component exists.

Statutory Deductions Validation

Some validation that's enforced via the user interface can't be enforced whilst loading records via HCM Data Loader. We would therefore advise that you run either the Payroll Validation Report or the Diagnostic report after loading Statutory Deduction records via HCM Data Loader.

Message	Resolution
An active calculation card of this type for this tax reporting unit already exists.	There can be only one effective Statutory Deductions calculation card related to a specific TRU on any date, for a payroll relationship.
An active calculation card of this type, with no associations, already exists.	There can be only one active Statutory Deductions card that's isn't associated to a TRU at any one time.
The priority period type is invalid for the combination of association details with this calculation card.	A priority period type is entered for the calculation component, but it doesn't correspond to one of the payroll frequencies assigned to one of the component associations.
You must enter a priority period type because there are association details with different payroll frequencies.	If there are multiple component associations with different payroll frequencies, the priority period type must be entered so that the payroll run can process these associations with that same frequency.
NI category {NI_CATEGORY} isn't valid for the current certificate.	The NI category is dependent upon the certificate entered on the Aggregation Information. Select a valid NI Category for the certificate.
Pension basis {PENSION_BASIS} isn't valid for the current NI category {NI_CATEGORY}.	The pension basis field is dependent upon the NI category.

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Statutory Deductions for the UK
- Guidelines for Updating Aggregation Information for UK Statutory Deductions
- Guidelines for Updating PAYE for UK Statutory Deductions
- Guidelines for Updating NI for UK Statutory Deductions
- Guidelines for Loading UK Statutory Deductions Card Associations

Guidelines for Loading Statutory Deductions for the UK

You need to have one Calculation Card record for every UK employee you are maintaining Statutory Deduction data for.

Even if you are updating an existing Statutory Deduction card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Statutory Deductions Calculation Card Attributes

The Statutory Deductions calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Statutory Deductions calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Statutory Deductions'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Examples of supplying these attributes are included in the example topics for Statutory Deductions.

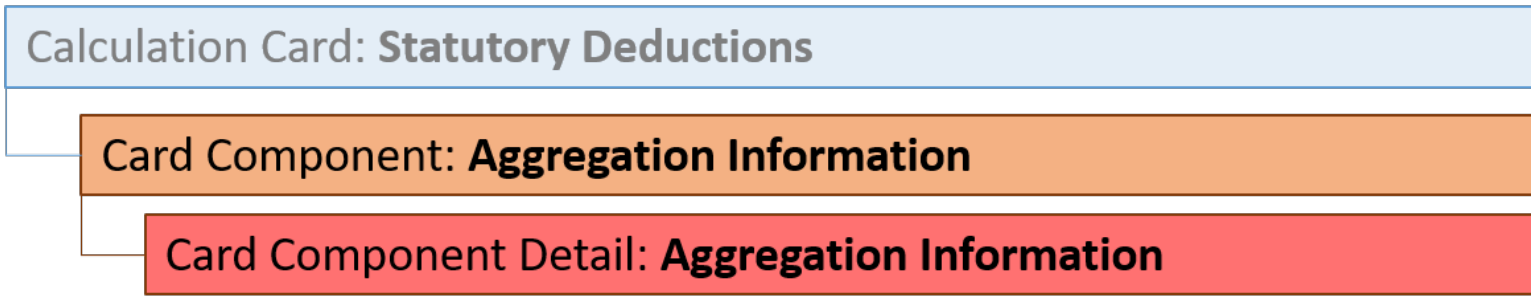
Related Topics

- Overview of Statutory Deductions for the UK
- Guidelines for Updating Aggregation Information for UK Statutory Deductions
- Guidelines for Updating PAYE for UK Statutory Deductions
- Guidelines for Updating NI for UK Statutory Deductions
- Guidelines for Loading UK Statutory Deductions Card Associations

Guidelines for Updating Aggregation Information for UK Statutory Deductions

The Aggregation Information card component captures common information across multiple NI categories. It is created for you if the Statutory Deductions card is autogenerated.

If you need to update the defaulted aggregation information, provide an Aggregation Information card component and related Component Detail for each tax reporting unit for the employee.



The Aggregation Information card component uses the Aggregation Information flexfield context to capture data. Flexfield segments are uploaded using the Component Detail record type.

Card Component Attributes for Aggregation Information

The Aggregation Information card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Aggregation Information card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions calculation card should be identified by using the same key type used to identify the calculation card.When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Aggregation Information card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Statutory Deductions calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Specify 'Aggregation Information'.
ComponentSequence	N/A	Specify '1'.

Card Component Detail Attributes for Aggregation Information

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context: Aggregation Information (HRX_GB_AGGREGATION_INFO)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Aggregation Information card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Aggregation Information card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_GB_AGGREGATION_INFO'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF		Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Statutory Deductions for the UK
- Guidelines for Loading Statutory Deductions for the UK
- Example of Updating Aggregation Information for UK Statutory Deductions
- Guidelines for Loading Flexfield Data

Example of Updating Aggregation Information for UK Statutory Deductions

This example updates the existing Aggregation Information card component for employee assignment E451388 on the 1st January 2018.

User keys are provided on the assumption that the Statutory Deductions card was autogenerated and therefore the source key values are unknown. Use the CalculationCard.dat file to upload the Aggregation Information card component information with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|UK LDG|Statutory Deductions|2018/01/01|E451388 |1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|UK LDG|Statutory Deductions|2018/01/01|E451388 |1|Aggregation Information|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|FLEX:Deduction Developer DF|DirInformationCategory|
_PROCESS_TYPE(Deduction Developer DF=HRX_GB_AGGREGATION_INFO)|_NUMBER_OF_PERIODS(Deduction Developer
 DF=HRX_GB_AGGREGATION_INFO)
MERGE|ComponentDetail|UK LDG|Statutory Deductions|2018/01/01|E451388 |1|Aggregation Information|1|
HRX_GB_AGGREGATION_INFO|HRX_GB_AGGREGATION_INFO|MP|1
```

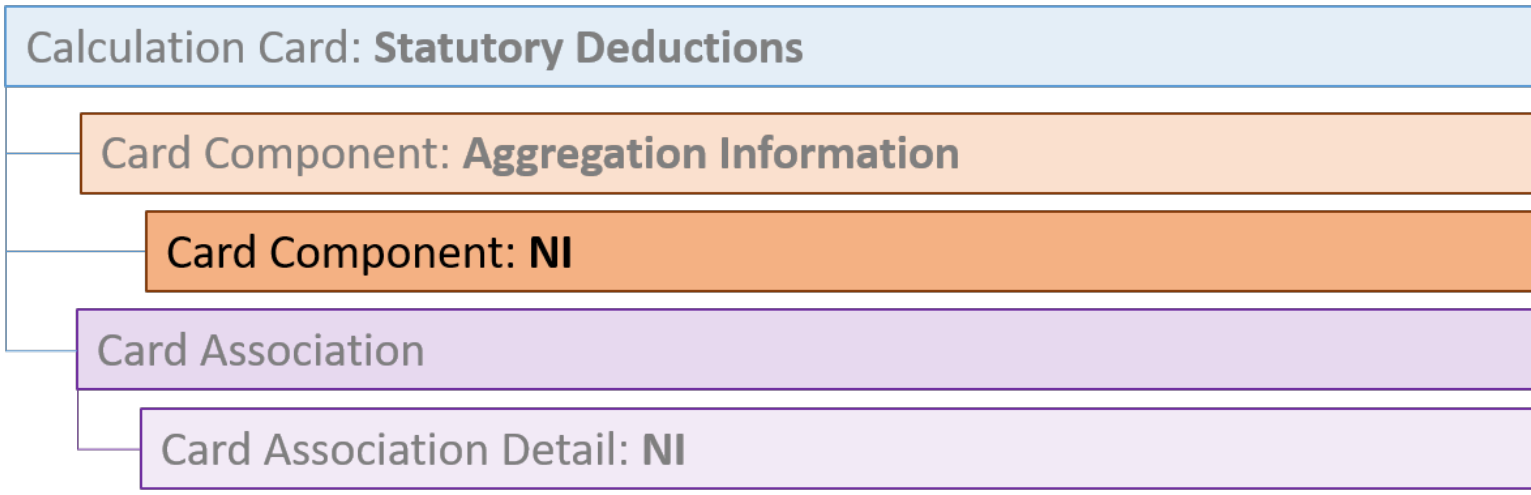
Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Statutory Deductions for the UK
- Guidelines for Loading Statutory Deductions for the UK
- Guidelines for Updating Aggregation Information for UK Statutory Deductions
- Guidelines for Loading Flexfield Data

Guidelines for Updating NI for UK Statutory Deductions

The NI card component captures data used to calculate National Insurance contributions.

It is created for you if the Statutory Deductions card is autogenerated. To update the NI information, you must provide a NI card component for each tax reporting unit the employee reports to.



The NI card component doesn't use flexfield contexts and so no Component Detail record is required. The NI card component references the Aggregation Information card component as its parent. The card association details reference the NI card component, so if creating a new NI component also create the NI card association detail record, refer to the Guidelines for Loading UK Statutory Deductions Card Associations topic.

Card Component Attributes for NI

The NI card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the NI card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
ParentDirCardCompId(Sour ceSystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	Identify the Aggregation Information card component for the employee. You can either provide this attribute with the SourceSystemId value supplied for the Aggregation Information card component, or provide the user key attributes.
EffectiveStartDate	N/A	The start date of the NI card component, typically the employee's start date. This must

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		be the same as the EffectiveStartDate on the Statutory Deductions calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Specify 'NI'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.
Context1	N/A	The NI Category. This is validated using the lookup type HRX_GB_NI_CATEGORY.
Context2	N/A	The pension basis. This is validated using the lookup type HRX_GB_PENSION_BASIS.

Related Topics

- Overview of Statutory Deductions for the UK
- Guidelines for Loading Statutory Deductions for the UK
- Example of Updating NI for UK Statutory Deductions

Example of Updating NI for UK Statutory Deductions

This example updates the existing NI card component for employee assignment E955165 on the 4th May 2022.

User keys are provided on the assumption that the Statutory Deductions card was auto generated and therefore the source key values are unknown. To create a complete Statutory Deductions card refer to the example at the end of the Statutory Deductions section. The CalculationCard.dat file is used to upload the NI card component information with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|UK LDG|Statutory Deductions|2022/05/04|E955165|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|Context1|Context2
MERGE|CardComponent|UK LDG|Statutory Deductions|2022/05/04|E955165|1|NI|1|C|N
```

Related Topics

- Overview of Statutory Deductions for the UK
- Guidelines for Loading Statutory Deductions for the UK
- Guidelines for Updating NI for UK Statutory Deductions
- Guidelines for Loading UK Statutory Deductions Card Associations

Guidelines for Updating PAYE for UK Statutory Deductions

The PAYE card component captures data used to calculate Tax contributions. It is created for you if the Statutory Deductions card is auto-generated.

If you are updating the PAYE data, you must provide a PAYE card component with a Component Detail record for each flexfield context relevant to the employee, for each tax reporting unit the employee reports to.

Card Component Attributes for PAYE

The PAYE card component uses the following attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the PAYE card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the PAYE card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Statutory Deductions calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Specify 'PAYE'.
ComponentSequence	N/A	The number to uniquely identify the PAYE card component when multiple PAYE card components exist. Not required when the source key is used to identify the card component.

Card Component Detail Attributes for PAYE

The PAYE card component uses four flexfield contexts. In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Pay As You Earn (HRX_GB_PAYE)

- New Starter Declaration (HRX_GB_NEW_STARTER)
- Pensioner Declaration (HRX_GB_PENSIONER)
- Expatriate Declaration (HRX_GB_EXPATRIATE)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent PAYE card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the PAYE card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_GB_PAYE'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Example of Updating PAYE for UK Statutory Deductions

This example updates the existing PAYE card component for employee assignment E8173668 on the 9th April 2022.

User keys are provided on the assumption that the Statutory Deductions card was auto generated and therefore the source key values are unknown. The CalculationCard.dat file is used to upload the PAYE card component information with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|UK LDG|Statutory Deductions|2022/04/09|E8173668|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|UK LDG|Statutory Deductions|2022/04/09|E8173668|1|PAYE|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|FLEX:Deduction Developer DF|DirInformationCategory|
_TAX_CODE(Deduction Developer DF=HRX_GB_PAYE)
MERGE|ComponentDetail|UK LDG|Statutory Deductions|2022/04/09|E8173668|1|PAYE|1|HRX_GB_PAYE|HRX_GB_PAYE|486L
```

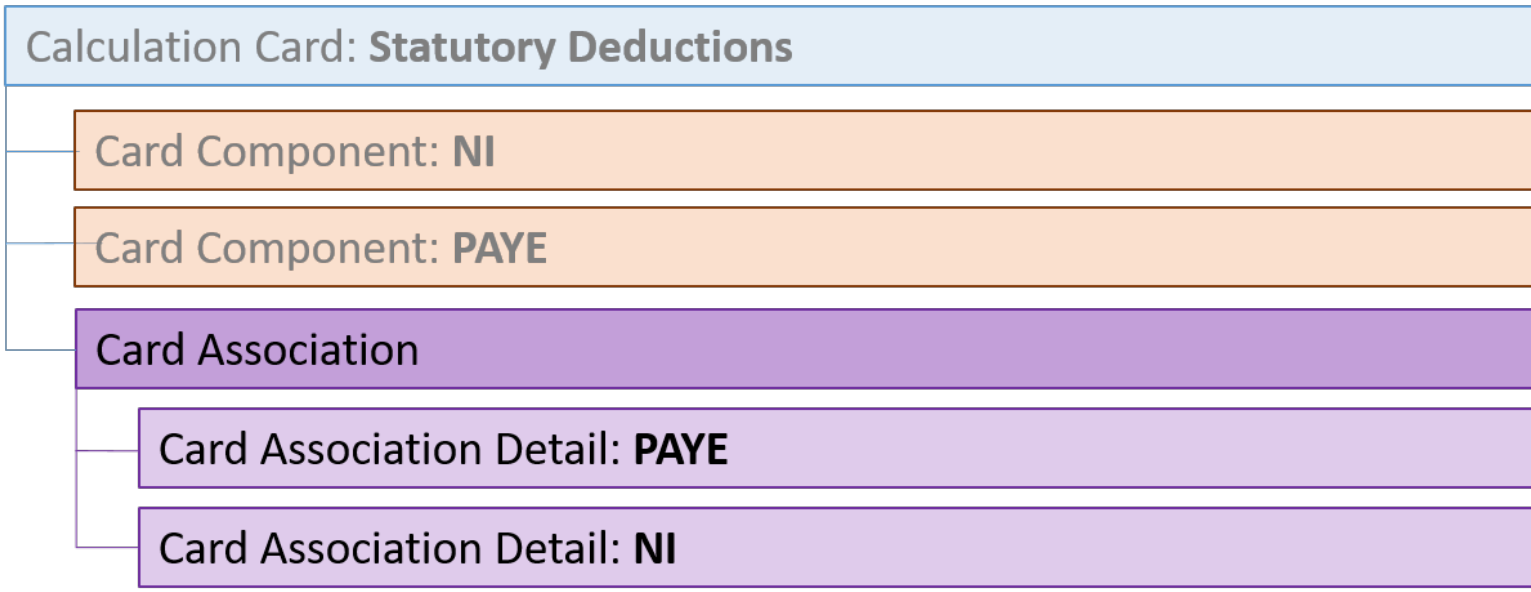
Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Statutory Deductions for the UK
- Guidelines for Loading Statutory Deductions for the UK
- Guidelines for Loading Flexfield Data

Guidelines for Loading UK Statutory Deductions Card Associations

The card association record associates the Statutory Deductions calculation card with a tax reporting unit.

If you provide the tax reporting unit during the new hire flow, the card association will be created automatically, else it must be created after the Statutory Deductions card is auto-generated. If creating a Statutory Deductions card from scratch you can provide the association details with the Statutory Deductions card.



The associated tax reporting unit is defined in the card association. The card association details records allow the PAYE and NI card components to be associated with the employee assignments.

Card Association Attributes for Statutory Deductions

The Card Association record type associates a Statutory Deductions card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Statutory Deductions card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Tax Withholding calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Tax Withholding calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.
TRUBatchId	N/A	The unique ID that provides an optional grouping mechanism for certain statutory reports. Use the lookup type previously created and associated to the TRU.

Card Association Detail Attributes for Statutory Deductions

The card association details record associates the NI and PAYE card component with the payroll assignments for the employee.

If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate		The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Identify the NI or PAYE card component this association is for.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of Statutory Deductions for the UK
- Guidelines for Loading Statutory Deductions for the UK
- Example of Loading UK Statutory Deduction Card Associations

Example of Loading UK Statutory Deduction Card Associations

This example creates the tax reporting unit association for the existing Statutory Deductions card belonging to employee assignment E451388.

User keys are provided to reference the existing Statutory Deductions card on the assumption it was auto generated and therefore the source key values are unknown. The CalculationCard.dat file is used to upload the Statutory Deductions card associations with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|UK LDG|Statutory Deductions|2015/04/16|E451388|1

METADATA|CardAssociation|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|TaxReportingUnitName|SourceSystemOwner|SourceSystemId
MERGE|CardAssociation|UK LDG|Statutory Deductions|2018/01/01|E451388|1|UK Tax Reporting Unit|VISION|
SD_451388_ASSOC

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|EffectiveStartDate|
DirRepCardId(SourceSystemId)|DirCardCompId(SourceSystemId)|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|SD_451388_ASSOC_NI|2018/01/01|SD_451388_ASSOC|SD_451388_NI|E451388
MERGE|CardAssociationDetail|VISION|SD_451388_ASSOC_PAYE|2018/01/01|SD_451388_ASSOC|SD_451388_PAYE|E451388
```

Related Topics

- Overview of Statutory Deductions for the UK
- Guidelines for Loading Statutory Deductions for the UK
- Guidelines for Updating Aggregation Information for UK Statutory Deductions
- Guidelines for Updating PAYE for UK Statutory Deductions
- Guidelines for Updating NI for UK Statutory Deductions
- Guidelines for Loading UK Statutory Deductions Card Associations

Example of Loading a New Starter Declaration for an Autogenerated UK Statutory Deductions Card

This example uploads the New Starter Declaration details for employee assignment E451388 on 1st January 2018 where the Statutory Deductions card already exists.

This example uploads the New Starter Declaration details for employee assignment E451388 on the 1st January 2018 where the Statutory Deductions card already exists.

Other examples can be found in the other Statutory Deduction topics within this section.

The CalculationCard.dat file is used to upload Statutory Deductions calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|UK LDG|Statutory Deductions|2018/01/01|E451388|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|UK LDG|Statutory Deductions|2018/01/01|E451388|1|PAYE|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|
AssignmentNumber|CardSequence|DirCardCompDefName|ComponentSequence|FLEX:Deduction Developer
 DF|DirInformationCategory|_EMP_DECLARED(Deduction Developer DF=HRX_GB_NEW_STARTER)|
_FINISH_STUDIES(Deduction Developer DF=HRX_GB_NEW_STARTER)|_NOT_PAID_BEFORE_TAX_YEAR(Deduction
 Developer DF=HRX_GB_NEW_STARTER)|_NS_STATEMENT(Deduction Developer DF=HRX_GB_NEW_STARTER)|
_RTI_FILED(Deduction Developer DF=HRX_GB_NEW_STARTER)|_STUDENT_LOAN_COMPANY_PAID(Deduction Developer
 DF=HRX_GB_NEW_STARTER)|_STUDENT_LOAN_DEDUCTIONS(Deduction Developer DF=HRX_GB_NEW_STARTER)|
_STUDENT_LOAN_NOT_FULL_PAID(Deduction Developer DF=HRX_GB_NEW_STARTER)|Pglnotpaid(Deduction Developer
 DF=HRX_GB_NEW_STARTER)|finishedPostgraduateStudiesBef(Deduction Developer DF=HRX_GB_NEW_STARTER)|
repayingPostgraduateLoanDirect(Deduction Developer DF=HRX_GB_NEW_STARTER)
MERGE|ComponentDetail|UK LDG|Statutory Deductions|2018/01/01|E451388|1|PAYE|1|HRX_GB_NEW_STARTER|
HRX_GB_NEW_STARTER|N|Y|N|A|N|Y|N|Y|N||
```

Related Topics

- Overview of Statutory Deductions for the UK
- Guidelines for Loading Statutory Deductions for the UK
- Guidelines for Updating PAYE for UK Statutory Deductions

Example of Creating a New UK Statutory Deductions Card

This example creates a complete Statutory Deductions card with associations for assignment E473381.

Other examples can be found in the other Statutory Deduction topics within this section.

The CalculationCard.dat file is used to upload Statutory Deductions calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|SD_451388|UK LDG|Statutory Deductions|2011/01/01|E473381

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|ParentDirCardCompId(SourceSystemId)|Context1|Context2
MERGE|CardComponent|VISION|SD_451388_AI|UK LDG|SD_451388|2011/01/01|Aggregation Information|||
MERGE|CardComponent|VISION|SD_451388_NI|UK LDG|SD_451388|2011/01/01|NI|SD_451388_AI|A|N
MERGE|CardComponent|VISION|SD_451388_PAYE|UK LDG|SD_451388|2011/01/01|PAYE|||

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer
 DF|DirInformationCategory|_PROCESS_TYPE(Deduction Developer DF=HRX_GB_AGGREGATION_INFO)|
_NUMBER_OF_PERIODS(Deduction Developer DF=HRX_GB_AGGREGATION_INFO)|_RENEWAL_DATE(Deduction Developer
 DF=HRX_GB_AGGREGATION_INFO)|_TAX_CODE(Deduction Developer DF=HRX_GB_PAYE)|_TAX_BASIS(Deduction
 Developer DF=HRX_GB_PAYE)|_PREVIOUS_TAXABLE_PAY_ON_P11(Deduction Developer DF=HRX_GB_PAYE)|
_PREVIOUS_TAX_PAID_ON_P11(Deduction Developer DF=HRX_GB_PAYE)|_AUTHORITY(Deduction Developer
 DF=HRX_GB_PAYE)|_PENSIONER(Deduction Developer DF=HRX_GB_PAYE)|_REASON_TO_WITHHOLD_TAX(Deduction
 Developer DF=HRX_GB_PAYE)|_P45_ACTION(Deduction Developer DF=HRX_GB_PAYE)|_P45_DATE_ISSUED(Deduction
 Developer DF=HRX_GB_PAYE)|_NUMBER_OF_PERIODS_CVD(Deduction Developer DF=HRX_GB_PAYE)|
_NUMBER_OF_HRS_WORKED(Deduction Developer DF=HRX_GB_PAYE)|_REPORT_NI_YTD_VALUES(Deduction Developer
 DF=HRX_GB_PAYE)|_IRREGULAR_EMP_PAYMENT(Deduction Developer DF=HRX_GB_PAYE)|_ONE_OFF_PAYMENT(Deduction
 Developer DF=HRX_GB_PAYE)|_PAYMENT_NON_INDIVIDUAL(Deduction Developer DF=HRX_GB_PAYE)|
```

```
_UNPAID_ABSENCE(Deduction Developer DF=HRX_GB_PAYE)|_ON_STRIKE(Deduction Developer DF=HRX_GB_PAYE)|
_SEND_HMRC_PAYROLL_ID_CHG(Deduction Developer DF=HRX_GB_PAYE)|_ORA_HRX_GB_LATE_PAYE_RSN(Deduction Developer
 DF=HRX_GB_PAYE)|_RETAIN_LATE_PAYE_RSN(Deduction Developer DF=HRX_GB_PAYE)|_FAPR(Deduction Developer
 DF=HRX_GB_PAYE)|_FAPR_CHECK(Deduction Developer DF=HRX_GB_PAYE)|_EXCLUDE_FROM_AL(Deduction Developer
 DF=HRX_GB_PAYE)|_SERIOUS_HEALTH_LUMP_SUM(Deduction Developer DF=HRX_GB_PAYE)
MERGE|ComponentDetail|VISION|SD_451388_AI_GBAI|UK LDG|SD_451388_AI|2011/01/01|Aggregation Information|
HRX_GB_AGGREGATION_INFO|HRX_GB_AGGREGATION_INFO|MP|2|||||||||||||||||||||||||
MERGE|ComponentDetail|VISION|SD_451388_PAYE_GBPAYE|UK LDG|SD_451388_PAYE|2011/01/01|PAYE|HRX_GB_PAYE|
HRX_GB_PAYE||||550L|C|45000|12000|P45|N|T|IAT||1||Y|N|N|N|N|N|N||N|N|N|N|N

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|SD_451388_ASSOC|UK LDG|SD_451388|2011/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirRepCardId(SourceSystemId)|EffectiveStartDate|DirCardCompId(SourceSystemId)|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|SD_451388_ASSOC_NI|UK LDG|SD_451388_ASSOC|2011/01/01|SD_451388_NI|E473381
MERGE|CardAssociationDetail|VISION|SD_451388_ASSOC_PAYE|UK LDG|SD_451388_ASSOC|2011/01/01|SD_451388_PAYE|
E473381
```

Taxable Benefits

Overview of Taxable Benefits for the UK

The Taxable Benefits card stores the employee's situation related to all benefits as part of their employment contracts, which are subjected to tax and national insurance.

The benefits provided to the employees can be processed through payroll or by P11D reporting. The employers can choose if they reports benefits using P11D or process them through payroll.

You can have a combination of the different types of benefit, either of P11D reporting or processing through payroll, provided the benefit type can be processed by payroll. Currently, HMRC regulations allow two benefit types to be reported only through P11D.

There may be instances where mass upload is required for the Taxable Benefits card:

During data migration:

- Employee benefits must be uploaded to ensure that all relevant information about Taxable Benefits is migrated from your legacy system.

Ongoing bulk updates:

- Bulk load of P11D data – Mass upload of P11D data can be done as part of the year-end processes to load employee benefits.

It is recommended to have a good understanding of the Taxable Benefits card and the information it contains prior to attempting mass upload. For further information, see Oracle Fusion HCM (United Kingdom): Processing Benefits-in-Kind and P11D Reporting (Doc ID 2393233.1)

Considerations and Prerequisites

The taxable benefits elements as well as the calculation card components for each of the taxable benefits are predefined. However you must create the element eligibility before the Taxable Benefit components are created for the employee.

The following is the list of predefined elements available and the eligibility requirements:

Element Name	Create Eligibility
Assets Transferred Processor	Yes
Assets Transferred Result	
Payments Made On Behalf Of Employee Processor	Yes
Payments Made On Behalf Of Employee Result	
Vouchers and Credit Cards Processor	Yes
Vouchers and Credit Cards Result	
Mileage Allowance And Passenger Payments Processor	Yes
Mileage Allowance And Passenger Payments Calculator	
Mileage Allowance And Passenger Payments Result	
Car And Car Fuel Processor	Yes
Car and Car Fuel Calculator	
Car and Car Fuel Result	
Private Medical Treatment Or Insurance Processor	Yes
Private Medical Treatment Or Insurance Result	
Qualifying Relocation Expenses Processor	Yes
Qualifying Relocation Expenses Calculator	
Qualifying Relocation Expenses Result	
Service Supplied Processor	Yes
Service Supplied Result	
Assets Placed At Employees Disposal Processor	Yes
Assets Placed At Employee Disposal Result	
Other Items Processor	Yes
Other Items Result	
Expenses Payments Processor	Yes
Expenses Payments Result	

All 'Processor' elements delivered require an eligibility created before it can be used.

Taxable Benefits Card Record Types

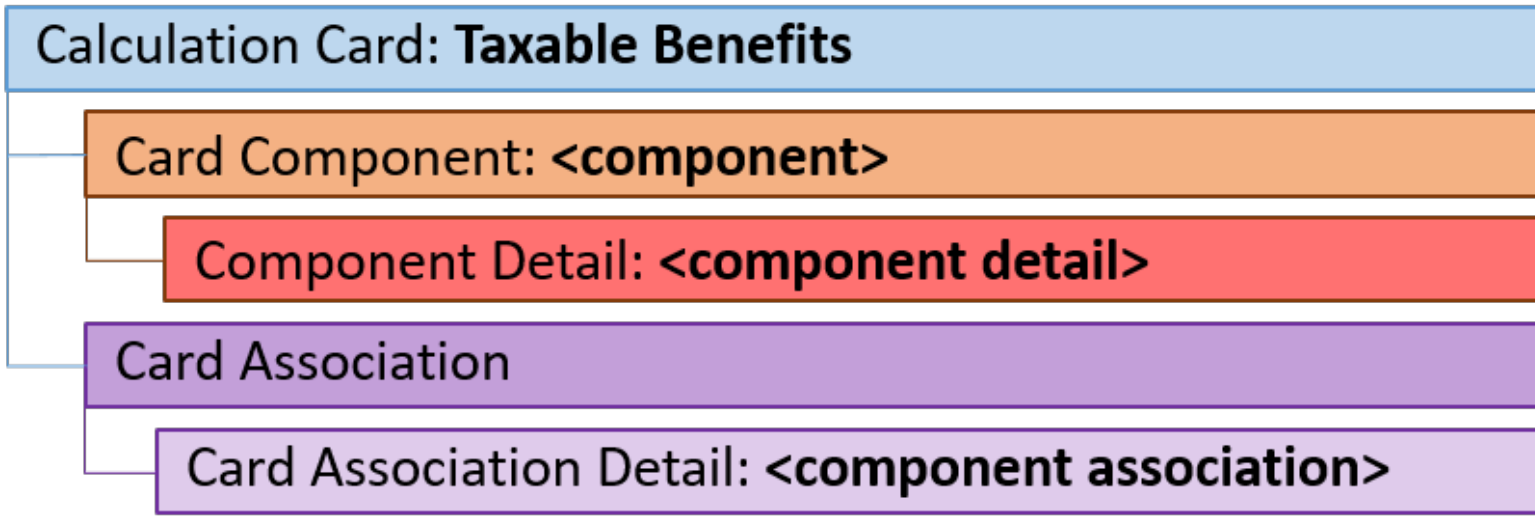
The Taxable Benefits card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements. UK Taxable Benefits utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Taxable Benefits Calculation Card Hierarchy

The Taxable Benefits calculation card supports several card components, though the typical Taxable Benefits card is likely to have just a few.

The generic shape of the Taxable Benefits calculation card is described in this diagram:



Some card components have multiple component details and child card components. Each top=level card component must be associated with the employee's assignments.

The shape that applies to each card component is described in the example topics within this section.

Guidelines for Loading Taxable Benefits for the UK

Taxable Benefits isn't an auto-generated card, if recording taxable benefit data for the first time. You need to create the Taxable Benefits card and its association with the employee's tax reporting unit.

Taxable Benefits Calculation Card Attributes

The Statutory Deductions calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Taxable Benefits calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Taxable Benefits'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card.
CardSeqeunce	N/A	Specify '1'.

These attributes are supplied against the CalculationCard discriminator. Always supply the CalculationCard record, even when updating an existing card.

Card Components

The following table lists the multiple card components applicable to the Taxable Benefits calculation card. Provide a card component record for each taxable benefits type required. The attributes to provide for each card component are common but some card components reference other card components as parents and when this is the case both card components should be supplied together.

Card Components Applicable to the Taxable Benefits Calculation Card

Card Component Name	Parent Card Component
Car and Car Fuel	N/A
Period of Unavailability Car	Car and Car Fuel
Expenses Payments on Behalf of Employee	N/A
Other Items	N/A
Assets Placed at Employee Disposal	N/A
Service Supplied	N/A
Qualifying Relocation Expenses and Benefits	N/A
Private Medical Treatment or Insurance	N/A
Interest Free and Low Interest Loans	N/A
Basic Mileage Details	N/A
Mileage Allowance Payments	Basic Mileage Details
Employer Providing Living Accommodation	N/A
Vouchers and Credit Cards	N/A
Assets Transferred	N/A
Payments Made on Behalf of Employee	N/A
Van and Van Fuel	N/A
Period of Unavailability Van	Van and Van Fuel

Card Component Attributes

When loading a new card component it may be that the Taxable Benefits calculation card already exists. Ensure that the new card component doesn't pre-date the Taxable Benefits card.

The various Taxable Benefits card components use these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Taxable Benefits calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
DirCardCompDefName	N/A	The component definition name. Refer to the table below for the list of card component names.
ParentDirCardCompId(SourceSystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	Identify the parent card component. You can either, provide this attribute with the SourceSystemId value supplied for the parent card component, or, provide the user key attributes. Only supply for card components that have a parent. Refer to the table below.
EffectiveStartDate	N/A	The start date of the card component. This must be the same as or after the EffectiveStartDate on the Taxable Benefits calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component.
ComponentSequence	N/A	
Context1	N/A	A mandatory value to identify the benefit. For example, for car it should be the car registration details.

These attributes are supplied against the CardComponent discriminator.

Component Details

Each Taxable Benefits card component utilizes one or more flexfield contexts. Provide a Component Detail record for each flexfield context within each component card, to load the flexfield segment values.

This table lists the flexfield contexts applicable for each card component:

Card Component Name	Flexfield Context
Car and Car Fuel	• Basic Car Details (ORA_HRX_GB_BSC_CAR_2017) • Car Processing Details (ORA_HRX_GB_CAR_PRC_2017) • Fuel Processing Details (ORA_HRX_GB_FUEL_PRC_2017)
Period of Unavailability Car	• Period Of Unavailability (ORA_HRX_GB_PERD_UNAVAIL_2017)

Card Component Name	Flexfield Context
Expenses Payments on Behalf of Employee	• Expenses Payments (ORA_HRX_GB_EXPS_PYMT_2017)
Other Items	• Other Items (ORA_HRX_GB_OTH_ITEM_2017)
Assets Placed at Employee Disposal	• Assets Placed at Employees Disposal (ORA_HRX_GB_ASSET_EMP_DISP_2017)
Service Supplied	• Basic Benefit Information (ORA_HRX_GB_BSC_BENEFIT_2017)
Qualifying Relocation Expenses and Benefits	• Qualifying Relocation Expenses (ORA_HRX_GB_QUAL_RL_EXPS_2017)
Private Medical Treatment or Insurance	• Basic Benefit Information (ORA_HRX_GB_BSC_BENEFIT_2017)
Interest Free and Low Interest Loans	• Interest Free and Low Interest Loans (ORA_HRX_GB_INTRST_LOAN_2017)
Basic Mileage Details	• Basic Mileage Details (ORA_HRX_GB_BSC_MILEAGE_2017)
Mileage Allowance Payments	• Mileage Allowance Payments (ORA_HRX_GB_MILEAGE_PASSNGR_2017)
Employer Providing Living Accommodation	• Basic Accommodation Details (ORA_HRX_GB_BSC_ACCOMD_2017) • Employer Provided Living Accommodation (ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)
Vouchers and Credit Cards	• Basic Benefit Information (ORA_HRX_GB_BSC_BENEFIT_2017)
Assets Transferred	• Assets Transferred (ORA_HRX_GB_ASSET_TRNFER_2017)
Payments Made on Behalf of Employee	• Payments Made on Behalf of Employee (ORA_HRX_GB_PYMT_BEHALF_EMP_2017)
Van and Van Fuel	• Fuel Processing Details (ORA_HRX_GB_FUEL_PRC_2017) • Van Processing Details (ORA_HRX_GB_VAN_PRC_2017)
Period of Unavailability Van	• Period Of Unavailability (ORA_HRX_GB_PERD_UNAVAIL_2017)

Note: You can find the flexfield segment attribute names for these flexfield contexts using the View Business Objects task.

Component Detail Attributes

These are the common attributes to supply for every component detail. The flexfield context will then determine the flexfield segment attributes to also include:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parentcard component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName		The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory		The code for the flexfield context, such as 'ORA_HRX_GB_BSC_CAR_2017'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF		Supply the same value as for the DirInformationCategory attribute.

These attributes are supplied against the ComponentDetail discriminator.

Card Association

The card association record associates the Taxable Benefits calculation card with the employee's tax reporting unit. When the employee has multiple tax reporting units, supply a card association for each tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Taxable Benefits card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	T The parent Taxable Benefits calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Tax Withholding calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Tax Withholding calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

These attributes are supplied against the CardAssociation discriminator.

Card Association Details

The card association details record associates the Taxable Benefits card component with the employee's assignments. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Identify the NI or PAYE card component this association is for.
RelationshipGroupId(SourceSystemId)	N/A	The surrogate ID of the payroll assignment to associate.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

These attributes are supplied against the CardAssociationDetail discriminator.

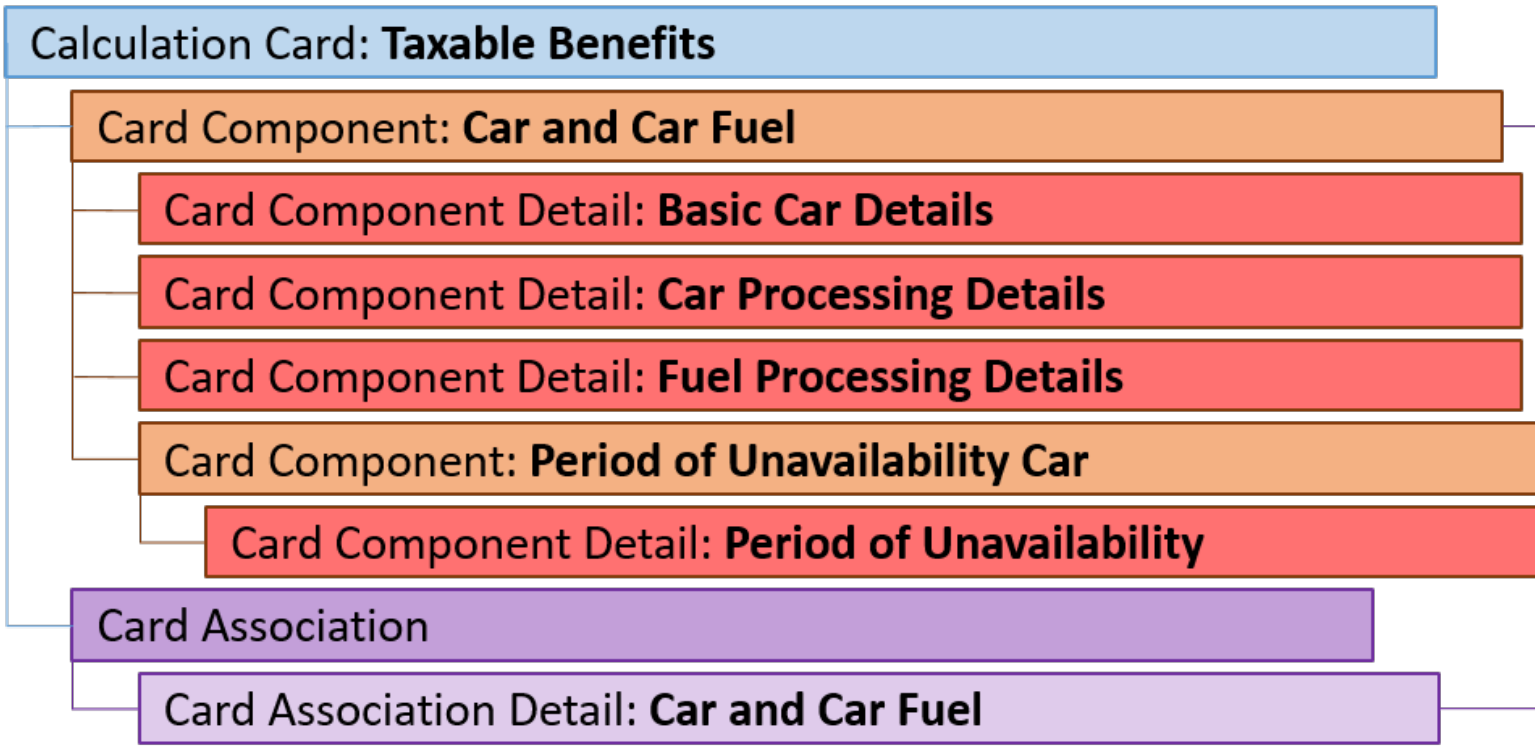
Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Flexfield Data

Example of Creating Car and Car Fuel Taxable Benefits for the UK

This example describes how you can create car and car fuel taxable benefits for the UK.

The Car and Car Fuel card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a Taxable Benefits card with the Car and Car Fuel and Period of Unavailability Car card components for employee assignment E451388 starting 6 April, 2020.

The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

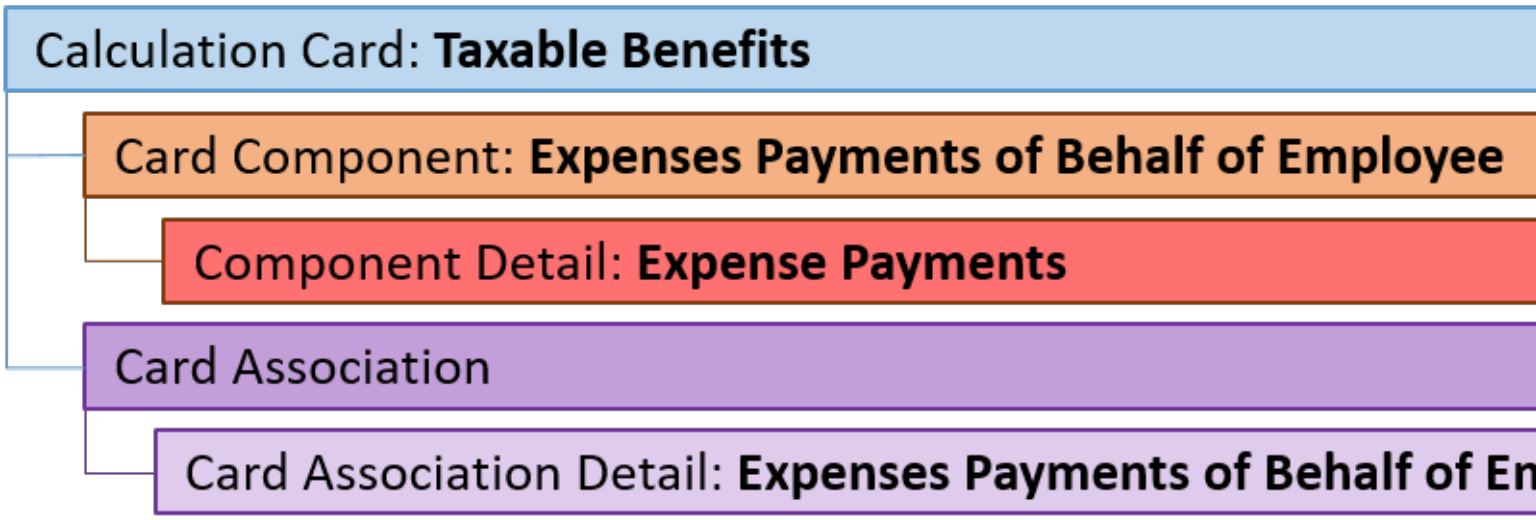
```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_451388|UK LDG|Taxable Benefits|2020/04/06|E451388

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|ParentDirCardCompId(SourceSystemId)|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|TB_451388_Car|UK LDG|TB_451388|2020/04/06||Car and Car Fuel|LF20 REG
MERGE|CardComponent|VISION|TB_451388_CUA|UK LDG|TB_451388|2020/04/06|TB_451388_Car|Period of Unavailability
 Car|

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction
 Developer DF|DirInformationCategory|_MAKE(Deduction Developer DF=ORA_HRX_GB_BSC_CAR_2017)|
_MODEL(Deduction Developer DF=ORA_HRX_GB_BSC_CAR_2017)|_DATE_FIRST_REGISTERED(Deduction Developer
 DF=ORA_HRX_GB_BSC_CAR_2017)|_LIST_PRICE(Deduction Developer DF=ORA_HRX_GB_BSC_CAR_2017)|
_APPROVED_CO2_EMISSION(Deduction Developer DF=ORA_HRX_GB_BSC_CAR_2017)|_CAR_FUEL_POWER(Deduction
 Developer DF=ORA_HRX_GB_BSC_CAR_2017)|_ENGINE_SIZE_CC(Deduction Developer DF=ORA_HRX_GB_BSC_CAR_2017)|
_ROTARY_ENGINE(Deduction Developer DF=ORA_HRX_GB_BSC_CAR_2017)|_PROCESS_OR_REPORT(Deduction Developer
 DF=ORA_HRX_GB_CAR_PRC_2017)|_PRICE_OF_ACCESSORIES(Deduction Developer DF=ORA_HRX_GB_CAR_PRC_2017)|
_CAPITAL_CONTRIBUTION(Deduction Developer DF=ORA_HRX_GB_CAR_PRC_2017)|_PAYMENT_MADE_PRIVATE(Deduction
 Developer DF=ORA_HRX_GB_CAR_PRC_2017)|_AMOUNT_FOREGONE(Deduction Developer DF=ORA_HRX_GB_CAR_PRC_2017)|
_CASH_EQUIVALENT(Deduction Developer DF=ORA_HRX_GB_CAR_PRC_2017)|_REGISTERED_ON_FPS(Deduction Developer
 DF=ORA_HRX_GB_CAR_PRC_2017)|_REGISTERED_ON_FPS_PA(Deduction Developer DF=ORA_HRX_GB_CAR_PRC_2017)|
_FUEL_BENEFIT_PROVIDED(Deduction Developer DF=ORA_HRX_GB_CAR_PRC_2017)|_DATE_FUEL_PROV_WITHDRAWN(Deduction
 Developer DF=ORA_HRX_GB_FUEL_PRC_2017)|_ADD_DAYS_AFT_WITHDRAWL(Deduction Developer
 DF=ORA_HRX_GB_FUEL_PRC_2017)|_FUEL_PROVISION_REINSTD(Deduction Developer DF=ORA_HRX_GB_FUEL_PRC_2017)|
```

```
_AMOUNT_FOREGONE(Deduction Developer DF=ORA_HRX_GB_FUEL_PRC_2017)|_CASH_EQUIVALENT(Deduction
 Developer DF=ORA_HRX_GB_FUEL_PRC_2017)|_ADDITION_NUM_DAYS_UNAVAIL(Deduction Developer
 DF=ORA_HRX_GB_PERD_UNAVAIL_2017)
MERGE|ComponentDetail|VISION|TB_451388_CAR2017|UK LDG|TB_451388_Car|2020/04/06|Car and Car Fuel|
ORA_HRX_GB_BSC_CAR_2017|ORA_HRX_GB_BSC_CAR_2017|Ford|Ka|2020/02/04|5999|100|A|1000|N|||||||||||||||
MERGE|ComponentDetail|VISION|TB_451388_PRC2017|UK LDG|TB_451388_Car|2020/04/06|Car and Car Fuel|
ORA_HRX_GB_CAR_PRC_2017|ORA_HRX_GB_CAR_PRC_2017|||||||||PAYROLL|3000|2500|2000|1500|1000|||N||||||
MERGE|ComponentDetail|VISION|TB_451388_FLP2017|UK LDG|TB_451388_Car|2020/04/06|Car and Car Fuel|
ORA_HRX_GB_FUEL_PRC_2017|ORA_HRX_GB_FUEL_PRC_2017|||||||||||||||||||2020/07/01|30|N|4000|3000|
MERGE|ComponentDetail|VISION|TB_451388_UNA2017|UK LDG|TB_451388_CUA|2020/04/06|Period of Unavailability Car|
ORA_HRX_GB_PERD_UNAVAIL_2017|ORA_HRX_GB_PERD_UNAVAIL_2017|||||||||||||||||||||||50

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_451388_TRU|UK LDG|TB_451388|2020/04/06|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_451388_ADC|TB_451388_Car|TB_451388_TRU|2020/04/06|E451388
```

Example of Creating Expense Payments on Behalf of Employee Taxable Benefits for the UK

This example describes how you can create expense payments on behalf of employee taxable benefits for the UK.

The Expense Payments on Behalf of Employee card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a Taxable Benefits card with the Expense Payments on Behalf of Employee card component for employee assignment E451351.

The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

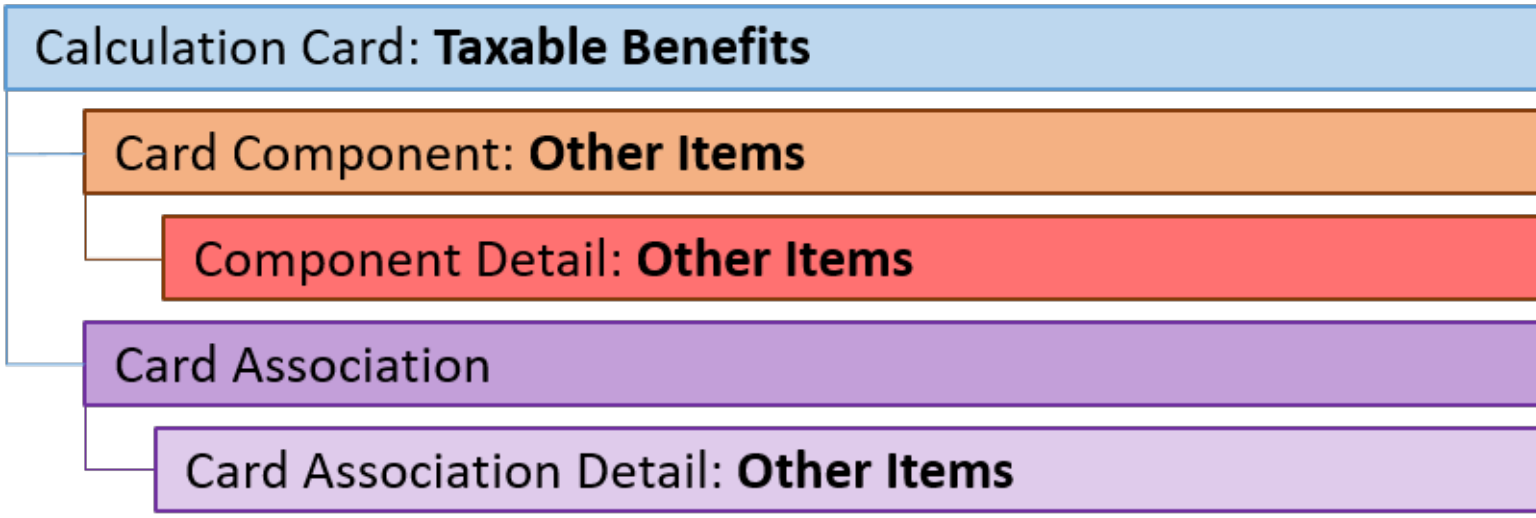
```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_451351|UK LDG|Taxable Benefits|2020/01/01|E451351
```

```
METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|TB_451351_EXPS|UK LDG|TB_451351|2020/01/01|Expenses Payments Made on Behalf of
 Employee|Exp Payment Ref

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer
 DF|DirInformationCategory|_PROCESS_OR_REPORT(Deduction Developer DF=ORA_HRX_GB_EXPS_PYMT_2017)|
_EXPENSE_PAYMENT_TYPE(Deduction Developer DF=ORA_HRX_GB_EXPS_PYMT_2017)|_COST_TO_EMPLOYER(Deduction
 Developer DF=ORA_HRX_GB_EXPS_PYMT_2017)|_AMOUNT_MADE_GOOD(Deduction Developer
 DF=ORA_HRX_GB_EXPS_PYMT_2017)|_DESC_OTHER_EXPENSES(Deduction Developer DF=ORA_HRX_GB_EXPS_PYMT_2017)|
_ENTERTAIN_EXPENSES_DISALLOW(Deduction Developer DF=ORA_HRX_GB_EXPS_PYMT_2017)|
_TAX_PROCESSING_RULE(Deduction Developer DF=ORA_HRX_GB_EXPS_PYMT_2017)
MERGE|ComponentDetail|VISION|TB_451351_EXPS_PYMT2017|UK LDG|TB_451351_EXPS|2020/01/01|Expenses Payments Made
 on Behalf of Employee|ORA_HRX_GB_EXPS_PYMT_2017|ORA_HRX_GB_EXPS_PYMT_2017|PAYROLL|OTHER_EXPENSES|9000|4000|
Freetext description|N|ACROSS_TAX_YEAR

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_451351_TRU|UK LDG|TB_451351|2020/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_451351_AD_EXP|TB_451351_EXPS|TB_451351_TRU|2020/01/01|E451351
```

Example of Creating Other Items Taxable Benefits for the UK

This example describes how you can create taxable benefits for other items for the UK.

The Other Items card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **Other Items** card component for employee assignment E453531.

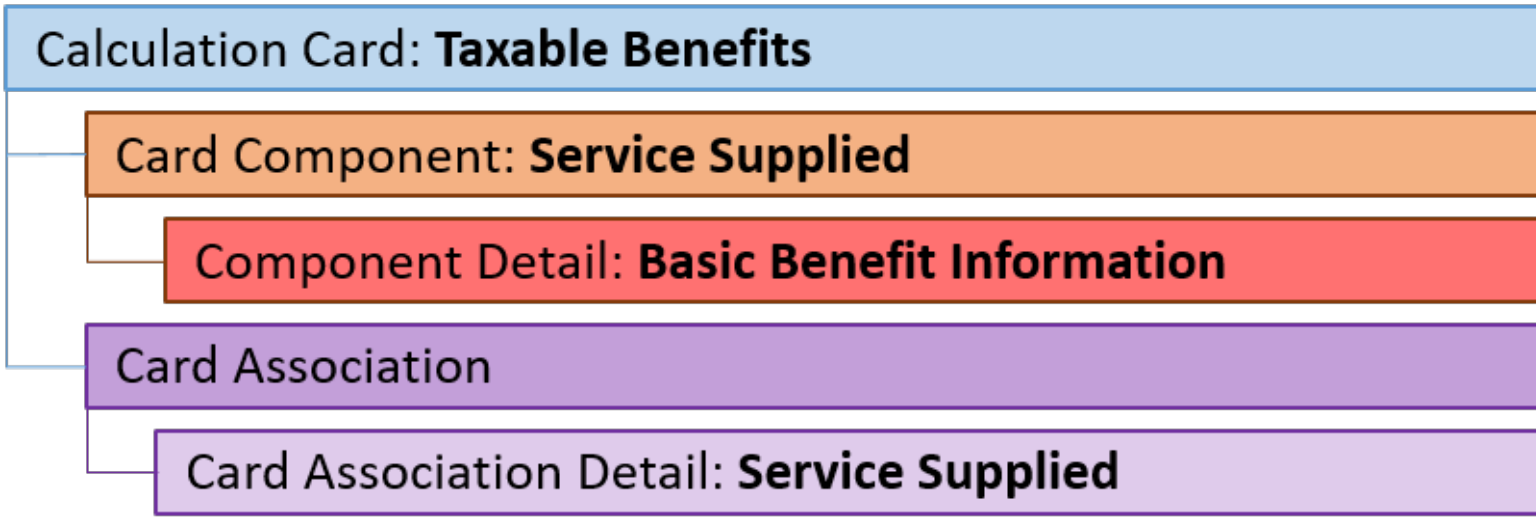
The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardDefinitionName|AssignmentNumber
MERGE|CalculationCard|VISION|TB_453531|UK LDG|2020/01/01|Taxable Benefits|E453531

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardId(SourceSystemId)|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|TB_453531_OI|UK LDG|2020/01/01|TB_453531|Other Items|BIKOTHITEM1

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardCompId(SourceSystemId)|DirCardCompDefName|FLEX:Deduction Developer DF|DirInformationCategory|
_PROCESS_OR_REPORT(Deduction Developer DF=ORA_HRX_GB_OTH_ITEM_2017)|_DESC_OTH_ITEMS_CLASS1A(Deduction
 Developer DF=ORA_HRX_GB_OTH_ITEM_2017)|_COST_TO_EMPLOYER_CLASS1A(Deduction Developer
 DF=ORA_HRX_GB_OTH_ITEM_2017)|_AMOUNT_MADE_GOOD_CLASS1A(Deduction Developer DF=ORA_HRX_GB_OTH_ITEM_2017)|
_DESC_OTH_ITEMS_NON_CLASS1A(Deduction Developer DF=ORA_HRX_GB_OTH_ITEM_2017)|_COST_TO_EMPLOYER(Deduction
 Developer DF=ORA_HRX_GB_OTH_ITEM_2017)|_AMOUNT_MADE_GOOD(Deduction Developer DF=ORA_HRX_GB_OTH_ITEM_2017)|
_INCOME_TAX_PAID(Deduction Developer DF=ORA_HRX_GB_OTH_ITEM_2017)|_SUBJECT_TO_TAX(Deduction Developer
 DF=ORA_HRX_GB_OTH_ITEM_2017)|_SUBJECT_TO_EMP_NI(Deduction Developer DF=ORA_HRX_GB_OTH_ITEM_2017)|
_TAX_PROCESSING_RULE(Deduction Developer DF=ORA_HRX_GB_OTH_ITEM_2017)
MERGE|ComponentDetail|VISION|TB_453531_OTHITEM2017|UK LDG|2020/01/01|TB_453531_OI|Other Items|
ORA_HRX_GB_OTH_ITEM_2017|ORA_HRX_GB_OTH_ITEM_2017|PAYROLL|OTHER|12000|5000|OTHER|24000|10000|5000|N|N|
ACROSS_TAX_YEAR

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardId(SourceSystemId)|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_453531_TRU|UK LDG|2020/01/01|TB_453531|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
EffectiveStartDate|DirRepCardId(SourceSystemId)|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_453531_AD_OI|TB_453531_OI|2020/01/01|TB_453531_TRU|E453531
```

Example of Creating Assets Placed at Employee Disposal Taxable Benefits for the UK

This example describes how you can create taxable benefits for assets placed at employee disposal, for the UK.

The Assets Placed at Employee Disposal card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **Assets Placed at Employee Disposal** card component for employee assignment E459241.

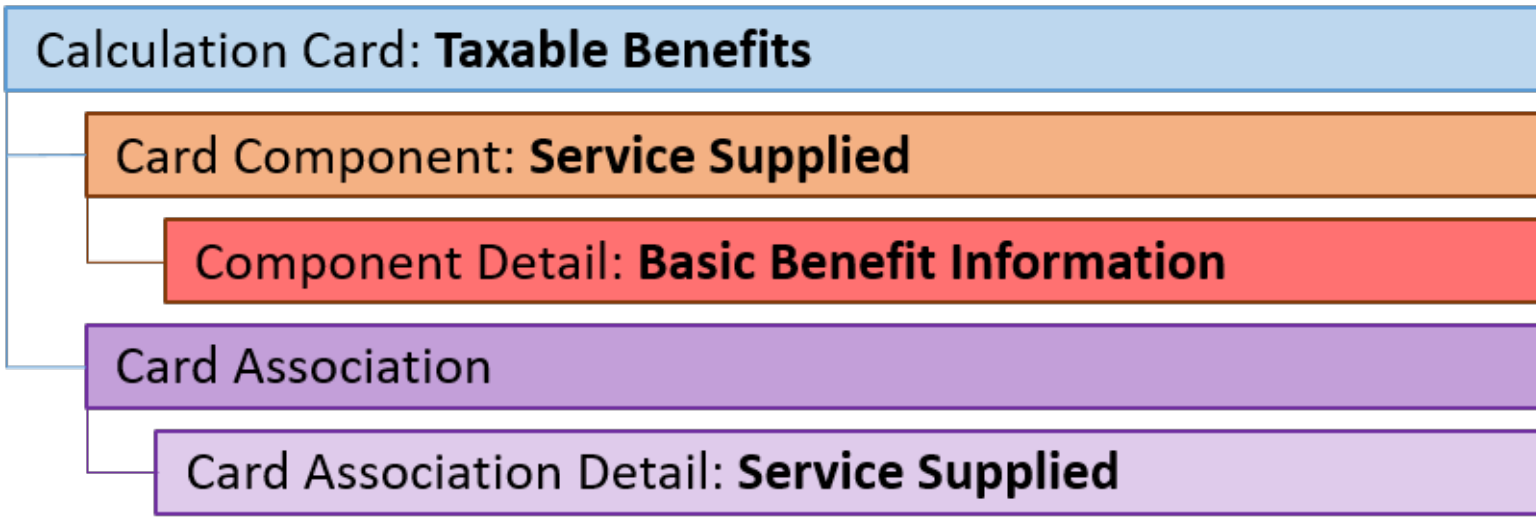
The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_459241|UK LDG|Taxable Benefits|2021/01/01|E459241

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|TB_459241_AEMPD|UK LDG|TB_459241|2021/01/01|Assets Placed at Employees Disposal|
Laptop

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer DF|
DirInformationCategory|_PROCESS_OR_REPORT(Deduction Developer DF=ORA_HRX_GB_ASSET_EMP_DISP_2017)|
_COST_TO_EMPLOYER(Deduction Developer DF=ORA_HRX_GB_ASSET_EMP_DISP_2017)|_ANNUAL_FLAG(Deduction
 Developer DF=ORA_HRX_GB_ASSET_EMP_DISP_2017)|_AMOUNT_MADE_GOOD(Deduction Developer
 DF=ORA_HRX_GB_ASSET_EMP_DISP_2017)|_OVERRIDING_DESC_FOR_P11D(Deduction Developer
 DF=ORA_HRX_GB_ASSET_EMP_DISP_2017)|_TAX_PROCESSING_RULE(Deduction Developer
 DF=ORA_HRX_GB_ASSET_EMP_DISP_2017)
MERGE|ComponentDetail|VISION|TB_459241_ASSET_EMP_DISP_2017|UK LDG|TB_459241_AEMPD|2021/01/01|Assets Placed
 at Employees Disposal|ORA_HRX_GB_ASSET_EMP_DISP_2017|ORA_HRX_GB_ASSET_EMP_DISP_2017|PAYROLL|3000|N|200||
ACROSS_TAX_YEAR

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_459241_TRU|UK LDG|TB_459241|2021/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_459241_AD_AEMPD|TB_459241_AEMPD|TB_459241_TRU|2021/01/01|E459241
```

Example of Creating Service Supplied Taxable Benefits for the UK

This example describes how you can create service supplied taxable benefits for the UK.

The Service Supplied card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits** for the United Kingdom topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **Service Supplied** card component for employee assignment E462367.

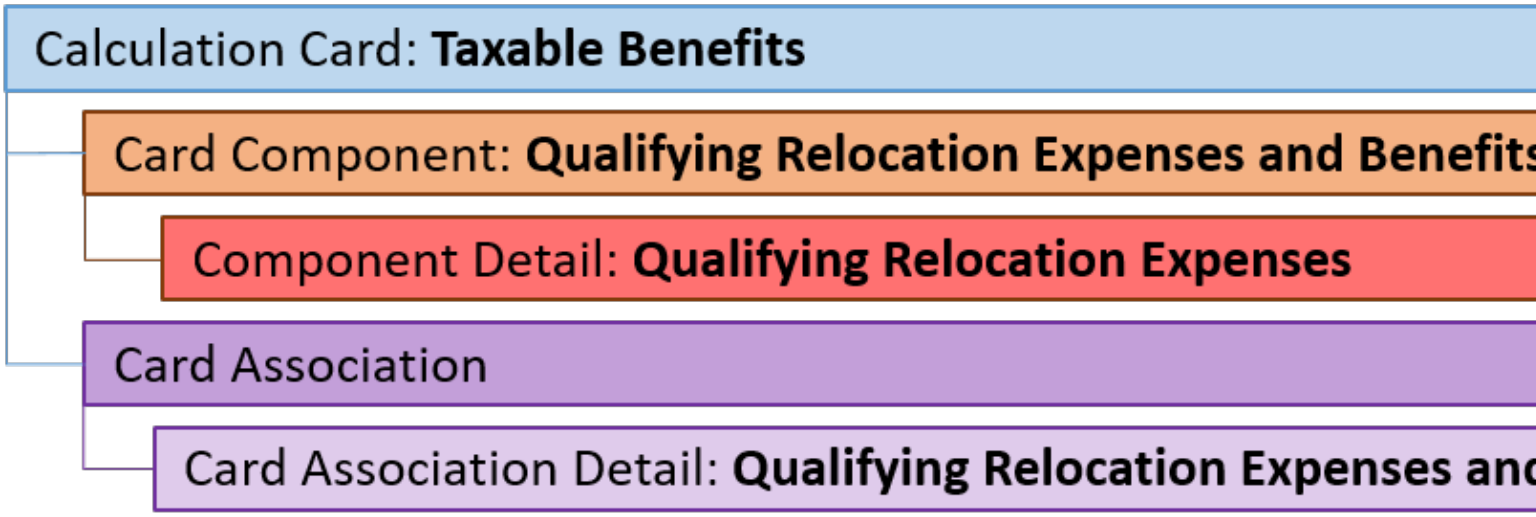
The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_462367|UK LDG|Taxable Benefits|2021/01/01|E462367

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|TB_462367_SERV|UK LDG|TB_462367|2021/01/01|Service Supplied|Serivce Supplied Ref

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer DF|
DirInformationCategory|_PROCESS_OR_REPORT(Deduction Developer DF=ORA_HRX_GB_BSC_BENEFIT_2017)|
_COST_TO_EMP_OR_VALUE(Deduction Developer DF=ORA_HRX_GB_BSC_BENEFIT_2017)|_ANNUAL_FLAG(Deduction Developer
 DF=ORA_HRX_GB_ASSET_EMP_DISP_2017)|_AMOUNT_MADE_GOOD(Deduction Developer DF=ORA_HRX_GB_BSC_BENEFIT_2017)|
_TAX_PROCESSING_RULE(Deduction Developer DF=ORA_HRX_GB_BSC_BENEFIT_2017)
MERGE|ComponentDetail|VISION|TB_462367_SERV_BEN2017|UK LDG|TB_462367_SERV|2021/01/01|Service Supplied|
ORA_HRX_GB_BSC_BENEFIT_2017|ORA_HRX_GB_BSC_BENEFIT_2017|PAYROLL|4000|N|400|ACROSS_TAX_YEAR

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_462367_TRU|UK LDG|TB_462367|2021/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_462367_AD_SERV|TB_462367_SERV|TB_462367_TRU|2021/01/01|E462367
```

Example of Creating Qualifying Relocation Expenses and Benefits Taxable Benefits for the UK

This example describes how you can create qualifying reloaction expenses and benefits taxable benefits for the UK.

The Qualifying Relocation Expenses and Benefits card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **Qualifying Relocation Expenses and Benefits** card component for employee assignment E456732.

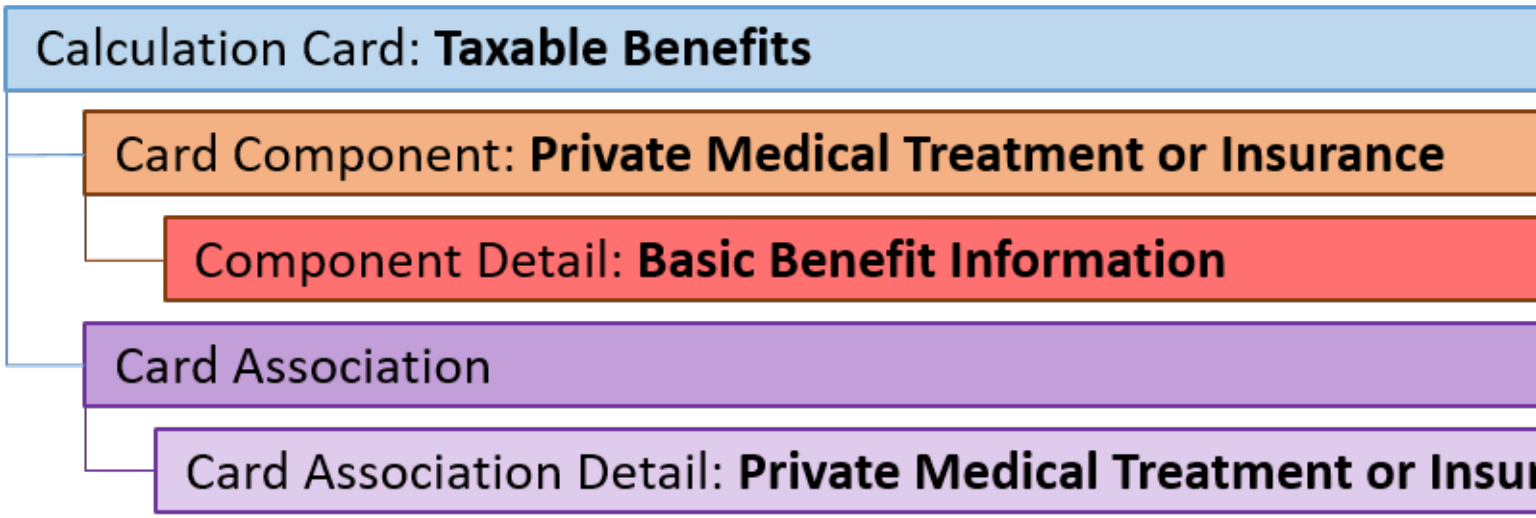
The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_456732|UK LDG|Taxable Benefits|2020/01/01|E456732

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|TB_456732_RELOC|UK LDG|TB_456732|2020/01/01|Qualifying Relocation Expenses and
 Benefits|Relocation Reference

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer DF|
DirInformationCategory|_PROCESS_OR_REPORT(Deduction Developer DF=ORA_HRX_GB_QUAL_RL_EXPS_2017)|
_GROSS_AMOUNT_QUAL_EXPENSES(Deduction Developer DF=ORA_HRX_GB_QUAL_RL_EXPS_2017)|
_COST_TO_EMPLOYER(Deduction Developer DF=ORA_HRX_GB_QUAL_RL_EXPS_2017)|_PAYMENT_MADE_EMPLOYEE(Deduction
 Developer DF=ORA_HRX_GB_QUAL_RL_EXPS_2017)|_COST_QUAL_LIVING_ACCOMD(Deduction
 Developer DF=ORA_HRX_GB_QUAL_RL_EXPS_2017)|_AMOUNT_QUAL_EXPENSES(Deduction Developer
 DF=ORA_HRX_GB_QUAL_RL_EXPS_2017)|_TAX_PROCESSING_RULE(Deduction Developer DF=ORA_HRX_GB_QUAL_RL_EXPS_2017)|
_CASH_EQUIVALENT(Deduction Developer DF=ORA_HRX_GB_QUAL_RL_EXPS_2017)
MERGE|ComponentDetail|VISION|TB_456732_RL_EXPS2017|UK LDG|TB_456732_RELOC|2020/01/01|Qualifying Relocation
 Expenses and Benefits|ORA_HRX_GB_QUAL_RL_EXPS_2017|ORA_HRX_GB_QUAL_RL_EXPS_2017|PAYROLL|7000|3000|2000|
1500|2500|ACROSS_TAX_YEAR|2000

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_456732_TRU|UK LDG|TB_456732|2020/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_456732_AD_RELOC|TB_456732_RELOC|TB_456732_TRU|2020/01/01|E456732
```

Example of Creating Private Medical Treatment or Insurance Taxable Benefits for the UK

This example describes how you can create taxable benefits for private medical treatment or insurance for the UK.

The Private Medical Treatment or Insurance card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **PrivateMedical Treatment or Insurance** card component for employee assignment E454923.

The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

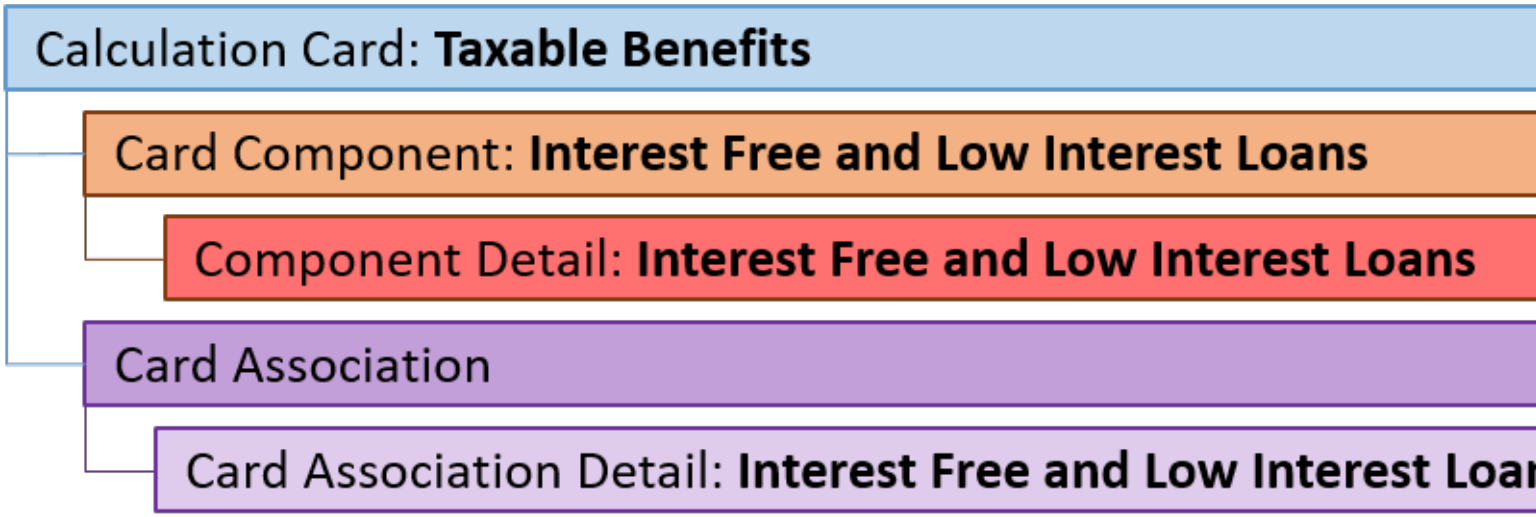
```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|EffectiveEndDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_454923|UK LDG|Taxable Benefits|2020/04/06||E454923

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|ComponentSequence|Context1
MERGE|CardComponent|VISION|TB_454923_MED|UK LDG|TB_454923|2020/04/06|2020/12/31|Private Medical Treatment or
 Insurance|1|Ref1
MERGE|CardComponent|VISION|TB_454923_MED|UK LDG|TB_454923|2021/01/01|2021/04/05|Private Medical Treatment or
 Insurance|1|Ref2

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|
FLEX:Deduction Developer DF|DirInformationCategory|_PROCESS_OR_REPORT_Display(Deduction Developer
 DF=ORA_HRX_GB_BSC_BENEFIT_2017)|_COST_TO_EMP_OR_VALUE(Deduction Developer DF=ORA_HRX_GB_BSC_BENEFIT_2017)|
_ANNUAL_FLAG(Deduction Developer DF=ORA_HRX_GB_BSC_BENEFIT_2017)|_AMOUNT_MADE_GOOD(Deduction Developer
 DF=ORA_HRX_GB_BSC_BENEFIT_2017)
MERGE|ComponentDetail|VISION|UKTAXBENBIK_COMPDTL_PRIVMED1|UK LDG|TB_454923_MED|2020/04/06|2020/12/31|Private
 Medical Treatment or Insurance|ORA_HRX_GB_BSC_BENEFIT_2017|ORA_HRX_GB_BSC_BENEFIT_2017|P11D reporting|5000|
Y|500
MERGE|ComponentDetail|VISION|UKTAXBENBIK_COMPDTL_PRIVMED1|UK LDG|TB_454923_MED|2021/01/01|2021/04/05|Private
 Medical Treatment or Insurance|ORA_HRX_GB_BSC_BENEFIT_2017|ORA_HRX_GB_BSC_BENEFIT_2017|P11D reporting|5000|
Y|500
```

```
METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_454923_TRU|UK LDG|TB_454923|2020/04/06||UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_454923_AD_MED|TB_454923_MED|TB_454923_TRU|2020/04/06||E454923
```

Example of Creating Interest Free and Low Interest Loans Taxable Benefits for the UK

This example describes how you can create taxable benefits for interest free and low interest loans for the UK.

The Interest Free and Low Interest Loans card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **Interest Free and Low Interest Loans** card component for employee assignment E472352.

The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

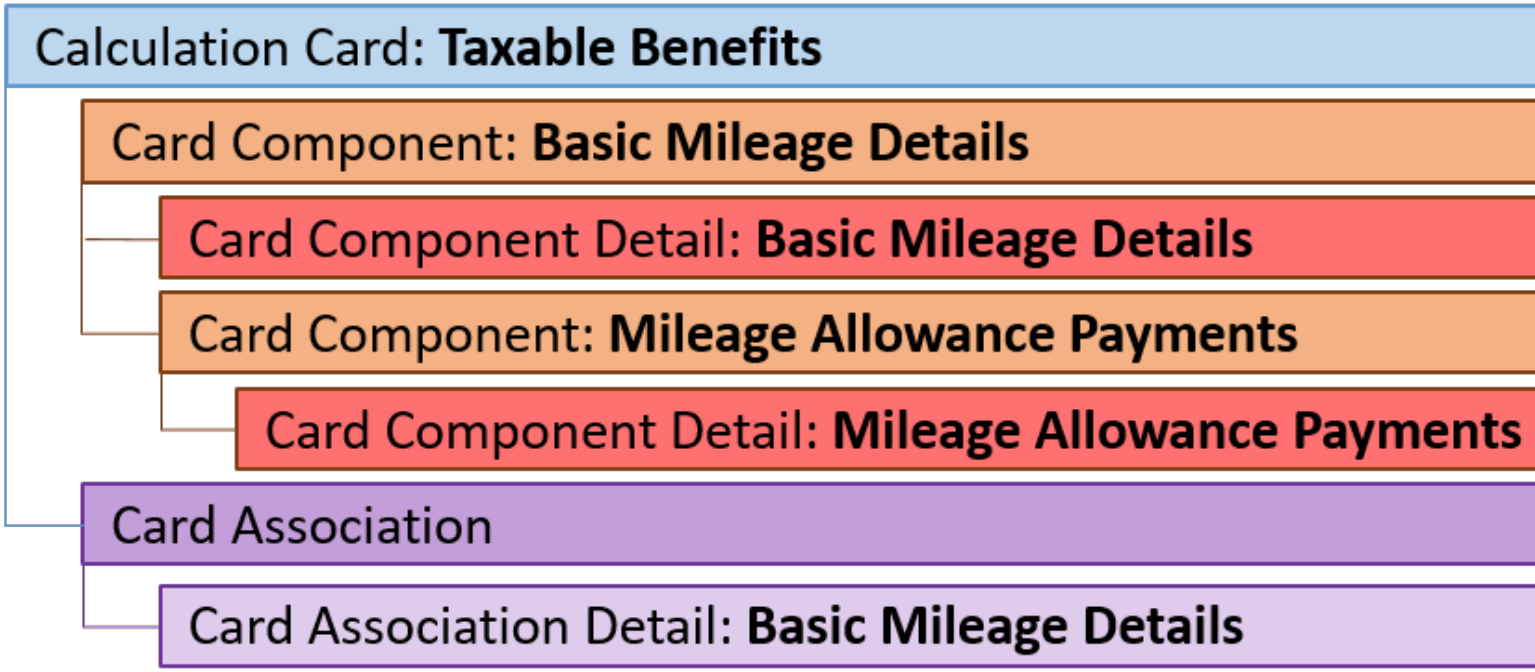
```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_472352|UK LDG|Taxable Benefits|2020/01/01|E472352

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|ComponentSequence|Context1
MERGE|CardComponent|VISION|TB_472352_LOAN|UK LDG|TB_472352|2020/01/01|Interest Free and Low Interest Loans|
1|Loan reference

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer DF|
DirInformationCategory|_PROCESS_OR_REPORT(Deduction Developer DF=ORA_HRX_GB_INTRST_LOAN_2017)|
_NUMBER_JNT_BORROWER(Deduction Developer DF=ORA_HRX_GB_INTRST_LOAN_2017)|_AMOUNT_OUTSTANDING_START(Deduction
 Developer DF=ORA_HRX_GB_INTRST_LOAN_2017)|_AMOUNT_OUTSTANDING_END(Deduction Developer
 DF=ORA_HRX_GB_INTRST_LOAN_2017)|_MAXIMUM_AMOUNT_OUTSTANDING(Deduction Developer
```

```
 DF=ORA_HRX_GB_INTRST_LOAN_2017)|_CURRENCY_OF_LOAN(Deduction Developer DF=ORA_HRX_GB_INTRST_LOAN_2017)|
_NUMBER_COMPLETE_TAX_MONTH(Deduction Developer DF=ORA_HRX_GB_INTRST_LOAN_2017)|_INTEREST_PAID_LOAN(Deduction
 Developer DF=ORA_HRX_GB_INTRST_LOAN_2017)|_CASH_EQUIVALENT(Deduction Developer
 DF=ORA_HRX_GB_INTRST_LOAN_2017)|_AMOUNT_FOREGONE(Deduction Developer DF=ORA_HRX_GB_INTRST_LOAN_2017)
MERGE|ComponentDetail|VISION|TB_472352_LOAN_2017|UK LDG|TB_472352_LOAN|2020/01/01|Interest Free and Low
 Interest Loans|ORA_HRX_GB_INTRST_LOAN_2017|ORA_HRX_GB_INTRST_LOAN_2017|P11D|2|5000|4000|2000||5|||

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_472352_TRU|UK LDG|TB_472352|2020/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_472352_AD_LOAN|TB_472352_LOAN|TB_472352_TRU|2020/01/01|E472352
```

Example of Creating Basic Mileage Details Taxable Benefits for the UK

This example describes how you can create taxable benefits for basic mileage details for the UK.

The Basic Mileage Details card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **Basic Mileage Details** and **Mileage Allowance Payments** card components for employee assignment E472317.

The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

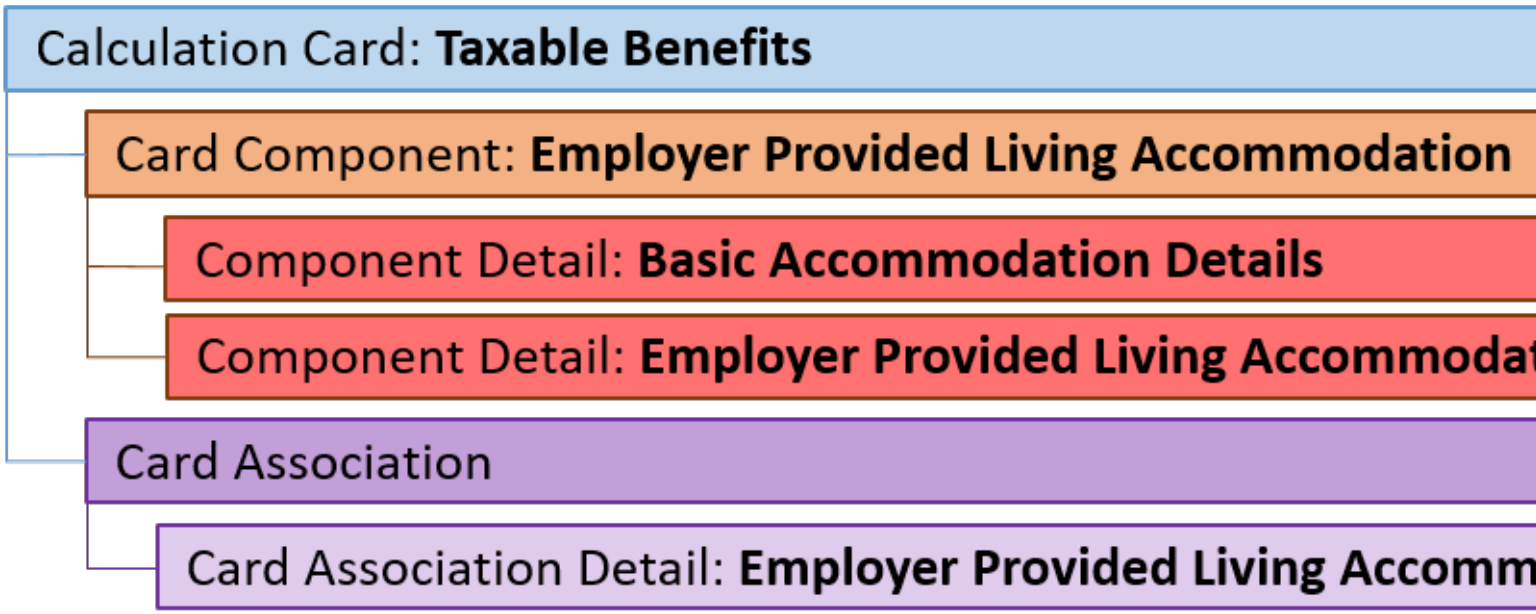
```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
```

```
MERGE|CalculationCard|VISION|TB_472317|UK LDG|Taxable Benefits|2020/01/01|E472317

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|ParentDirCardCompId(SourceSystemId)|Context1
MERGE|CardComponent|VISION|TB_472317_BM|UK LDG|TB_472317|2020/01/01|Basic Mileage Details||Mileage Reference
MERGE|CardComponent|VISION|TB_472317_MA|UK LDG|TB_472317|2020/01/01|Mileage Allowance Payments|TB_472317_BM|

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer DF|
DirInformationCategory|_PROCESS_OR_REPORT(Deduction Developer DF=ORA_HRX_GB_BSC_MILEAGE_2017)|
_VEHICLE_USED(Deduction Developer DF=ORA_HRX_GB_BSC_MILEAGE_2017)|_OPT_REMUNERATION_ARRANGE(Deduction
 Developer DF=ORA_HRX_GB_BSC_MILEAGE_2017)|_CASH_EQUIVALENT(Deduction Developer
 DF=ORA_HRX_GB_BSC_MILEAGE_2017)|_BUSINESS_MILES(Deduction Developer DF=ORA_HRX_GB_MILEAGE_PASSNGR_2017)|
_BUSINESS_MILES_NOT_REIMB(Deduction Developer DF=ORA_HRX_GB_MILEAGE_PASSNGR_2017)|_MILEAGE_PAYMENT(Deduction
 Developer DF=ORA_HRX_GB_MILEAGE_PASSNGR_2017)
MERGE|ComponentDetail|VISION|TB_472317_BM_MILEAGE_2017|UK LDG|TB_472317_BM|2020/01/01|Basic Mileage Details|
ORA_HRX_GB_BSC_MILEAGE_2017|ORA_HRX_GB_BSC_MILEAGE_2017|PAYROLL|CAR_OR_VAN|N|15000|||
MERGE|ComponentDetail|VISION|TB_472317_MA_PASSNGR_2017|UK LDG|TB_472317_MA|2020/01/01|Mileage Allowance
 Payments|ORA_HRX_GB_MILEAGE_PASSNGR_2017|ORA_HRX_GB_MILEAGE_PASSNGR_2017|||||500|250|4000

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_472317_TRU|UK LDG|TB_472317|2020/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_472317_AD_BASMIL|TB_472317_BM|TB_472317_TRU|2020/01/01|E472317
```

Example of Creating Employer Provided Living Accommodation Taxable Benefits for the UK

This example describes how you can create taxable benefits for employer provided living accommodation for the UK.

The Employer Provided Living Accommodation card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **Employer Provided Living Accommodation** card component for employee assignment E461672.

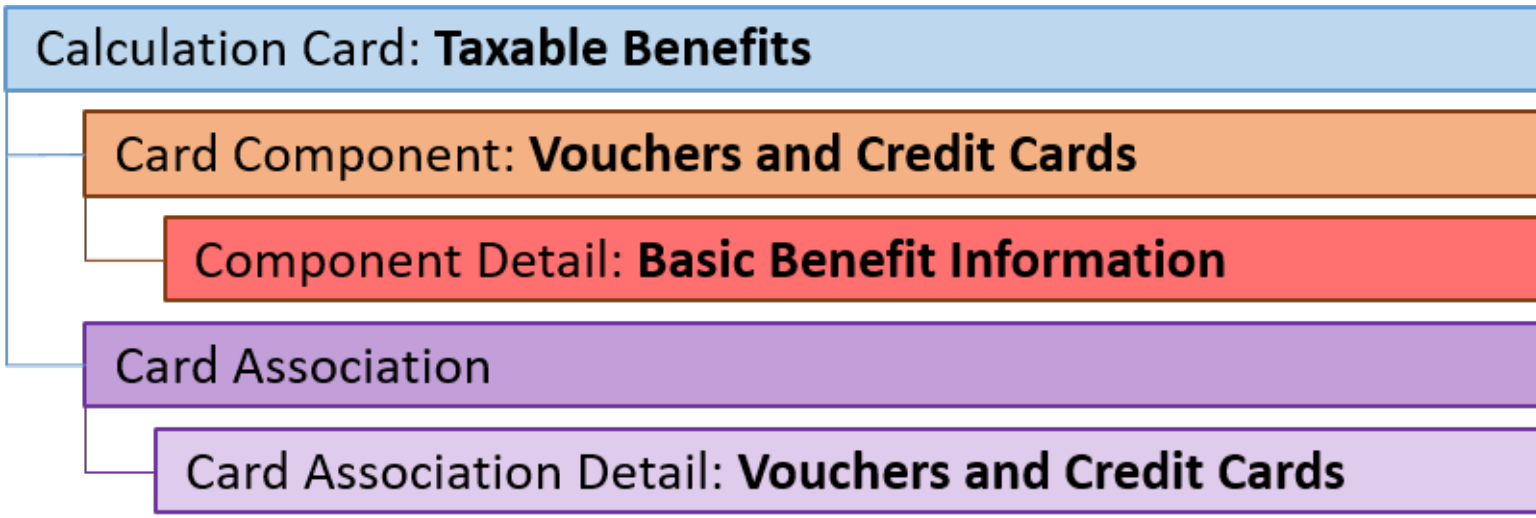
The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_461672|UK LDG|Taxable Benefits|2020/01/01|E461672

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|TB_461672_EPLA|UK LDG|TB_461672|2020/01/01|Employer Provided Living
 Accommodation|Reference

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer
 DF|DirInformationCategory|_ADDRESS_LINE1(Deduction Developer DF=ORA_HRX_GB_BSC_ACCOMD_2017)|
_ADDRESS_LINE2(Deduction Developer DF=ORA_HRX_GB_BSC_ACCOMD_2017)|_TOWN_OR_CITY(Deduction Developer
 DF=ORA_HRX_GB_BSC_ACCOMD_2017)|_POSTCODE(Deduction Developer DF=ORA_HRX_GB_BSC_ACCOMD_2017)|
_COUNTRY(Deduction Developer DF=ORA_HRX_GB_BSC_ACCOMD_2017)|_PROCESS_OR_REPORT(Deduction
 Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_RENTAL_VALUE(Deduction Developer
 DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_ANNUAL_VALUE(Deduction Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|
_AMOUNT_MADE_GOOD(Deduction Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|
_AMOUNT_FOREGONE(Deduction Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_CASH_EQUIVALENT(Deduction
 Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_COST_OF_ACCOMD(Deduction Developer
 DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_PAYMENT_MADE(Deduction Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|
_NUMBER_OF_DAYS(Deduction Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_RENT_PAID_BY_EMP(Deduction
 Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_OTHER_RENT_INCLUDED(Deduction
 Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_EXPENSES_INCURRED(Deduction Developer
 DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_HEATING(Deduction Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|
_LIGHTING(Deduction Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_REPAIR_DECORATION(Deduction
 Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_FURNITURE_GIVEN_TRNF(Deduction Developer
 DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_ANNUAL_VALUE_FURNITURE(Deduction Developer
 DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)|_OTHER(Deduction Developer DF=ORA_HRX_GB_EMPR_LIV_ACCOMD_2017)
MERGE|ComponentDetail|VISION|TB_461672_EPLA_BA2017|UK LDG|TB_461672_EPLA|2020/01/01|Employer Provided Living
 Accommodation|ORA_HRX_GB_BSC_ACCOMD_2017|ORA_HRX_GB_BSC_ACCOMD_2017|6 Test Road|Test Area|Reading|RG6 1RA|
GB|||||||||||||||||
MERGE|ComponentDetail|VISION|TB_461672_EPLA_LA2017|UK LDG|TB_461672_EPLA|2020/01/01|Employer Provided Living
 Accommodation|ORA_HRX_GB_EMPR_LIV_ACCOMD_2017|ORA_HRX_GB_EMPR_LIV_ACCOMD_2017||||||P11D|8000|10000|2000|
4000|3000|2500|1000|50|2500|500|N|N|N|N|N|N|Freetext Details

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_461672_TRU|UK LDG|TB_461672|2020/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_461672_AD_EPLA|TB_461672_EPLA|TB_461672_TRU|2020/01/01|E461672
```

Example of Creating Vouchers and Credit Cards Taxable Benefits for the UK

This example describes how you can create taxable benefits for vouchers ans credit cards for the UK.

The Vouchers and Credit Cards card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **Vouchers and Credit Cards** card component for employee assignment E442637.

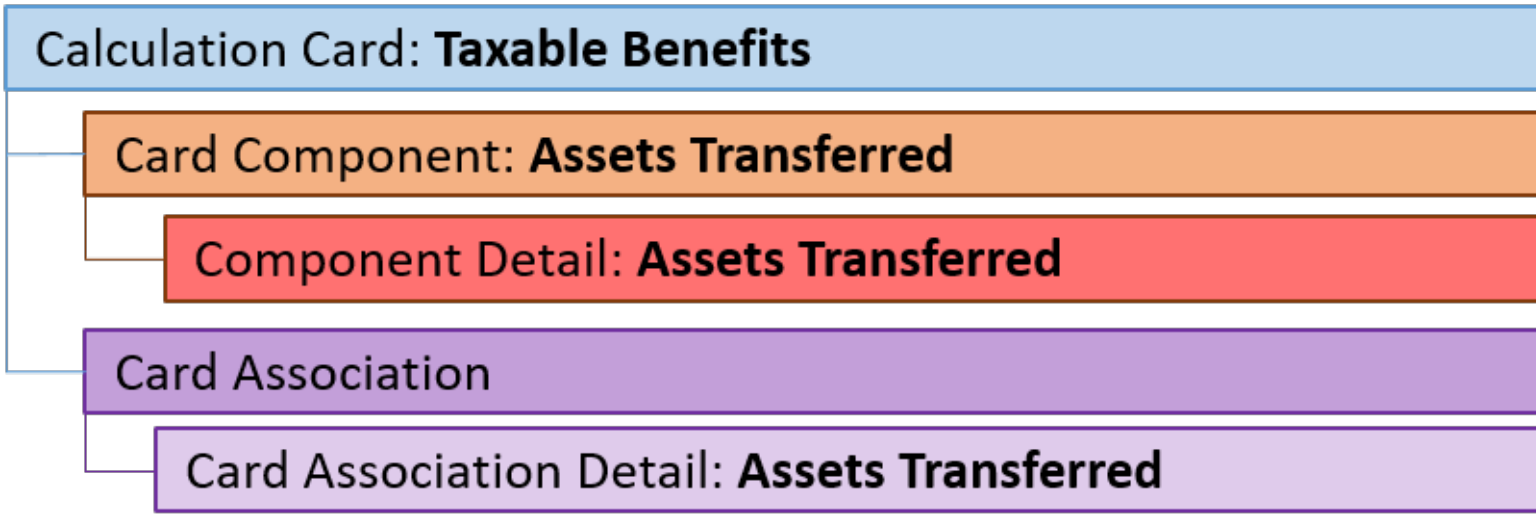
The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_442637|UK LDG|Taxable Benefits|2020/01/01|E442637

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|TB_442637_VCC|UK LDG|TB_442637|2020/01/01|Vouchers and Credit Cards|Voucher or
 Credit Card reference

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer DF|
DirInformationCategory|_PROCESS_OR_REPORT(Deduction Developer DF=ORA_HRX_GB_BSC_BENEFIT_2017)|
_COST_TO_EMP_OR_VALUE(Deduction Developer DF=ORA_HRX_GB_BSC_BENEFIT_2017)|_AMOUNT_MADE_GOOD(Deduction
 Developer DF=ORA_HRX_GB_BSC_BENEFIT_2017)|_TAX_PROCESSING_RULE(Deduction Developer
 DF=ORA_HRX_GB_BSC_BENEFIT_2017)
MERGE|ComponentDetail|VISION|TB_442637_VCC_BEN2017|UK LDG|TB_442637_VCC|2020/01/01|Vouchers and Credit
 Cards|ORA_HRX_GB_BSC_BENEFIT_2017|ORA_HRX_GB_BSC_BENEFIT_2017|PAYROLL|2000|100|ACROSS_TAX_YEAR

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_442637_TRU|UK LDG|TB_442637|2020/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_442637_AD_VCC|TB_442637_VCC|TB_442637_TRU|2020/01/01|E442637
```

Example of Creating Assets Transferred Taxable Benefits for the UK

This example describes how you can create taxable benefits for transferred assets for the UK.

The Assets Transferred card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **Assets Transferred** card component for employee assignment E442687.

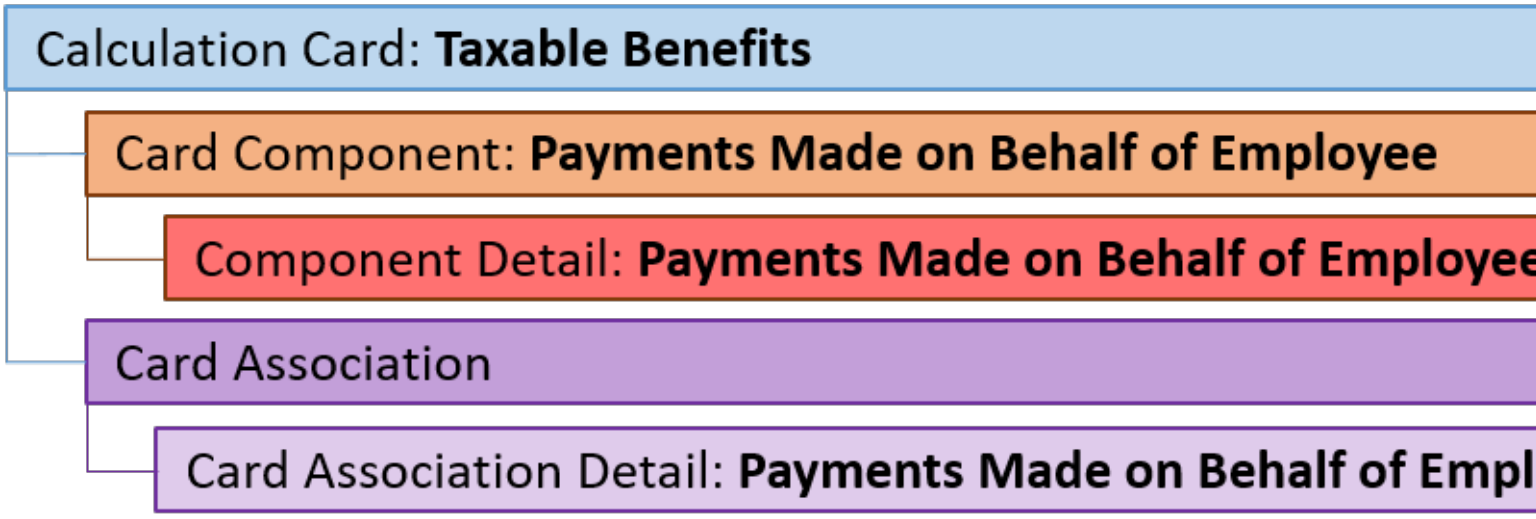
The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_442687|UK LDG|Taxable Benefits|2020/01/01|E442687

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|TB_442687_ASSET|UK LDG|TB_442687|2020/01/01|Assets Transferred|Assest reference

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer DF|
DirInformationCategory|_PROCESS_OR_REPORT(Deduction Developer DF=ORA_HRX_GB_ASSET_TRNFER_2017)|
_COST_TO_EMP_OR_VALUE(Deduction Developer DF=ORA_HRX_GB_ASSET_TRNFER_2017)|_AMOUNT_MADE_GOOD(Deduction
 Developer DF=ORA_HRX_GB_ASSET_TRNFER_2017)|_OVERRIDING_DESC_FOR_P11D(Deduction Developer
 DF=ORA_HRX_GB_ASSET_TRNFER_2017)|_TAX_PROCESSING_RULE(Deduction Developer DF=ORA_HRX_GB_ASSET_TRNFER_2017)
MERGE|ComponentDetail|VISION|TB_442687_ASSET_TRNFER_2017|UK LDG|TB_442687_ASSET|2020/01/01|Assets
 Transferred|ORA_HRX_GB_ASSET_TRNFER_2017|ORA_HRX_GB_ASSET_TRNFER_2017|PAYROLL|2000|100||ACROSS_TAX_YEAR

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_442687_TRU|UK LDG|TB_442687|2020/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_442687_AD_ASSET|TB_442687_ASSET|TB_442687_TRU|2020/01/01|E442687
```

Example of Creating Payments Made on Behalf of Employee Taxable Benefits for the UK

This example describes how you can create taxable benefits for payments made on behalf of employees, for the UK.

The Payments Made on Behalf of Employee card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **Payments Made on Behalf of Employee** card component for employee assignment E462367.

The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

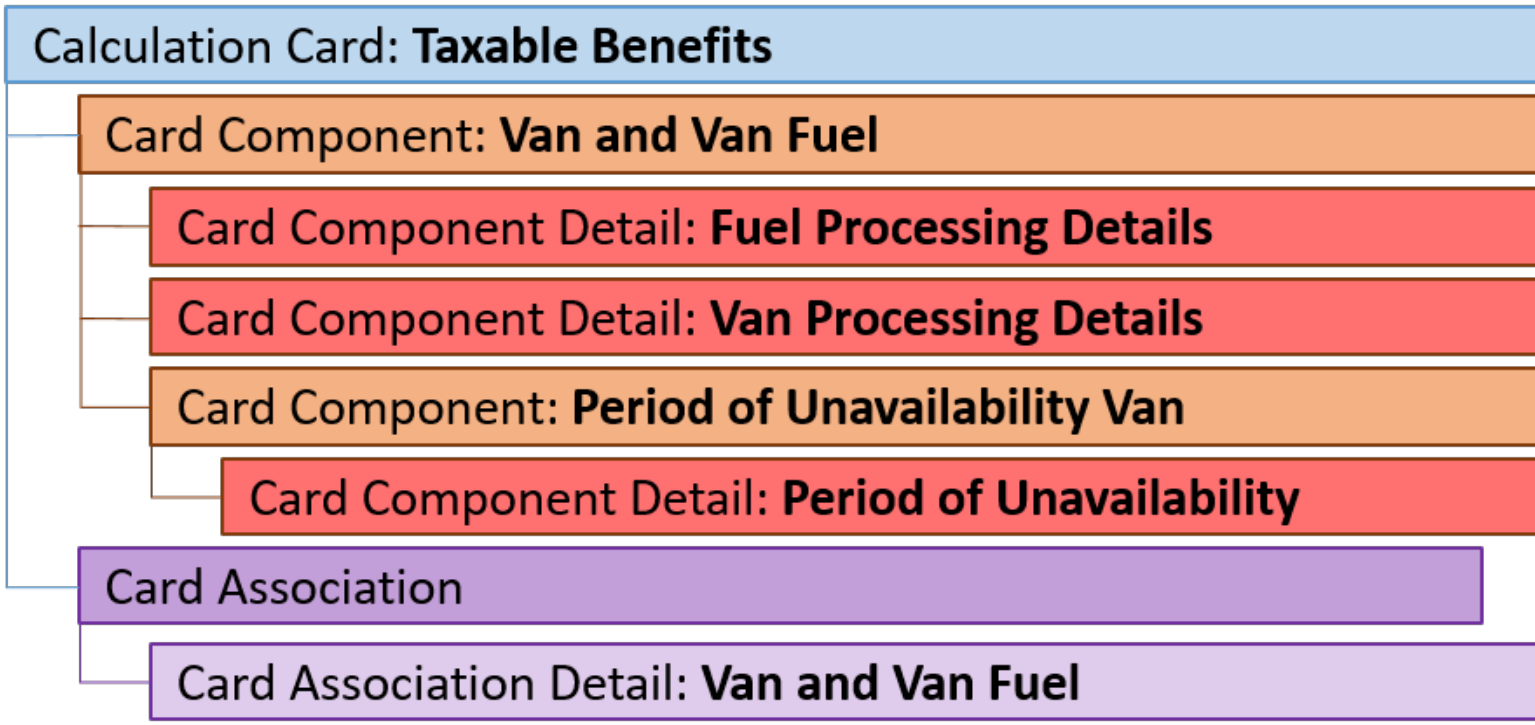
```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_462367|UK LDG|Taxable Benefits|2020/01/01|E462367

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|TB_462367_PAY4EMP|UK LDG|TB_462367|2020/01/01|Payments Made on Behalf of
 Employee|Payment reference

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer DF|
DirInformationCategory|_PROCESS_OR_REPORT(Deduction Developer DF=ORA_HRX_GB_PYMT_BEHALF_EMP_2017)|
_PAYMENT_AMOUNT(Deduction Developer DF=ORA_HRX_GB_PYMT_BEHALF_EMP_2017)|_TAX_ON_NOTION_PAYMENT(Deduction
 Developer DF=ORA_HRX_GB_PYMT_BEHALF_EMP_2017)|_OVERRIDING_DESC_FOR_P11D(Deduction
 Developer DF=ORA_HRX_GB_PYMT_BEHALF_EMP_2017)|_TAX_PROCESSING_RULE(Deduction Developer
 DF=ORA_HRX_GB_PYMT_BEHALF_EMP_2017)
MERGE|ComponentDetail|VISION|TB_462367_PAY4EMP_BEHALF_EMP_2017|UK LDG|TB_462367_PAY4EMP|2020/01/01|Payments
 Made on Behalf of Employee|ORA_HRX_GB_PYMT_BEHALF_EMP_2017|ORA_HRX_GB_PYMT_BEHALF_EMP_2017|PAYROLL|6000|
500||ACROSS_TAX_YEAR

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_462367_TRU|UK LDG|TB_462367|2020/01/01|UK TRU
```

```
METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_462367_AD_PAY4EMP|TB_462367_PAY4EMP|TB_462367_TRU|2020/01/01|E462367
```

Example of Creating Van and Van Fuel Taxable Benefits for the UK

This example describes how you can create taxable benefits for van and van fuel for the UK.

The Van and Van Fuel card component for the UK Taxable Benefits card has this shape:



Refer to the **Guidelines for Loading Taxable Benefits for the United Kingdom** topic for the attributes to supply for each record type.

This example creates a **Taxable Benefits** card with the **Van and Van Fuel** and **Period of Unavailability Van** card components for employee assignment E458323.

The CalculationCard.dat file is used to bulk-load Taxable Benefits cards with HCM Data Loader:

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|TB_458323|UK LDG|Taxable Benefits|2020/01/01|E458323

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|ParentDirCardCompId(SourceSystemId)|Context1
MERGE|CardComponent|VISION|TB_458323_VAN|UK LDG|TB_458323|2020/01/01|Van and Van Fuel||FL23 VAN
MERGE|CardComponent|VISION|TB_458323_VAN_AVAIL|UK LDG|TB_458323|2020/01/01|Period of Unavailability Van|
TB_458323_VAN|
```

```
METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|FLEX:Deduction Developer
 DF|DirInformationCategory|_PROCESS_OR_REPORT(Deduction Developer DF=ORA_HRX_GB_VAN_PRC_2017)|
_ZERO_EMISSION(Deduction Developer DF=ORA_HRX_GB_VAN_PRC_2017)|_PERCENT_REDUCTION_SHARING(Deduction
 Developer DF=ORA_HRX_GB_VAN_PRC_2017)|_EXPLANATION_SHARING(Deduction Developer DF=ORA_HRX_GB_VAN_PRC_2017)|
_PAYMENT_MADE_PRIVATE(Deduction Developer DF=ORA_HRX_GB_VAN_PRC_2017)|_AMOUNT_FOREGONE(Deduction
 Developer DF=ORA_HRX_GB_VAN_PRC_2017)|_CASH_EQUIVALENT(Deduction Developer DF=ORA_HRX_GB_VAN_PRC_2017)|
_FUEL_BENEFIT_PROVIDED(Deduction Developer DF=ORA_HRX_GB_VAN_PRC_2017)|_DATE_FUEL_PROV_WITHDRAWN(Deduction
 Developer DF=ORA_HRX_GB_FUEL_PRC_2017)|_ADD_DAYS_AFT_WITHDRAWL(Deduction Developer
 DF=ORA_HRX_GB_FUEL_PRC_2017)|_FUEL_PROVISION_REINSTD(Deduction Developer DF=ORA_HRX_GB_FUEL_PRC_2017)|
_AMOUNT_FOREGONE(Deduction Developer DF=ORA_HRX_GB_FUEL_PRC_2017)|_CASH_EQUIVALENT(Deduction
 Developer DF=ORA_HRX_GB_FUEL_PRC_2017)|_ADDITION_NUM_DAYS_UNAVAIL(Deduction Developer
 DF=ORA_HRX_GB_PERD_UNAVAIL_2017)
MERGE|ComponentDetail|VISION|TB_458323_VAN_VAN_PRC_2017|UK LDG|TB_458323_VAN|2020/01/01|Van and Van Fuel|
ORA_HRX_GB_VAN_PRC_2017|ORA_HRX_GB_VAN_PRC_2017|PAYROLL|N|10|Freetext explanation|2000|3000|4000|N||||||
MERGE|ComponentDetail|VISION|TB_458323_VAN_FUEL_PRC_2017|UK LDG|TB_458323_VAN|2020/01/01|Van and Van Fuel|
ORA_HRX_GB_FUEL_PRC_2017|ORA_HRX_GB_FUEL_PRC_2017|||||||||2020/07/01|20|N|3000|2000|
MERGE|ComponentDetail|VISION|TB_458323_VAN_UNAVAIL_2017|UK LDG|TB_458323_VAN_AVAIL|2020/01/01|Period of
 Unavailability Van|ORA_HRX_GB_PERD_UNAVAIL_2017|ORA_HRX_GB_PERD_UNAVAIL_2017|||||||||||||60

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|TB_458323_TRU|UK LDG|TB_458323|2020/01/01|UK TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirCardCompId(SourceSystemId)|
DirRepCardId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|TB_458323_AD_VAN|TB_458323_VAN|TB_458323_TRU|2020/01/01|E458323
```

24 Loading Payroll Localization Data for the United States

Benefits and Pensions 403 (b) and 457 (b)

Overview of Benefits and Pensions 403 (b) and 457 (b) for the US

Deferred compensation plans allow you to save for retirement on a pretax (or after-tax) basis.

Considerations and Prerequisites

You must define your 403 (b) or 457 (b) elements, prior to assigning a calculation card to employees you want to enroll in the plans. This card defines the plan the employee is enrolled in and the process for withholding the deductions (percentage or flat amount).

The payroll process observes these maximum contribution limits, as they are defined by the Internal Revenue Service (IRS):

- Deferred Compensation 403 (b) Annual Limit
- Deferred Compensation 457 (b) Annual Limit
- Deferred Compensation 403 (b) Catch-Up Annual Limit
- Deferred Compensation 457 (b) Catch-Up Annual Limit

You can override eligible compensation annual limit at the organization level.

> **Note:** When you create a 403 (b) or a 457 (b) deduction element, the payroll process doesn't take the deduction until you feed the eligible balance. The Elements task doesn't automatically establish this feed for tracking eligible earnings. You must feed these balances for your eligible earnings results element.

Benefits and Pensions 403 (b) and 457 (b) Record Types

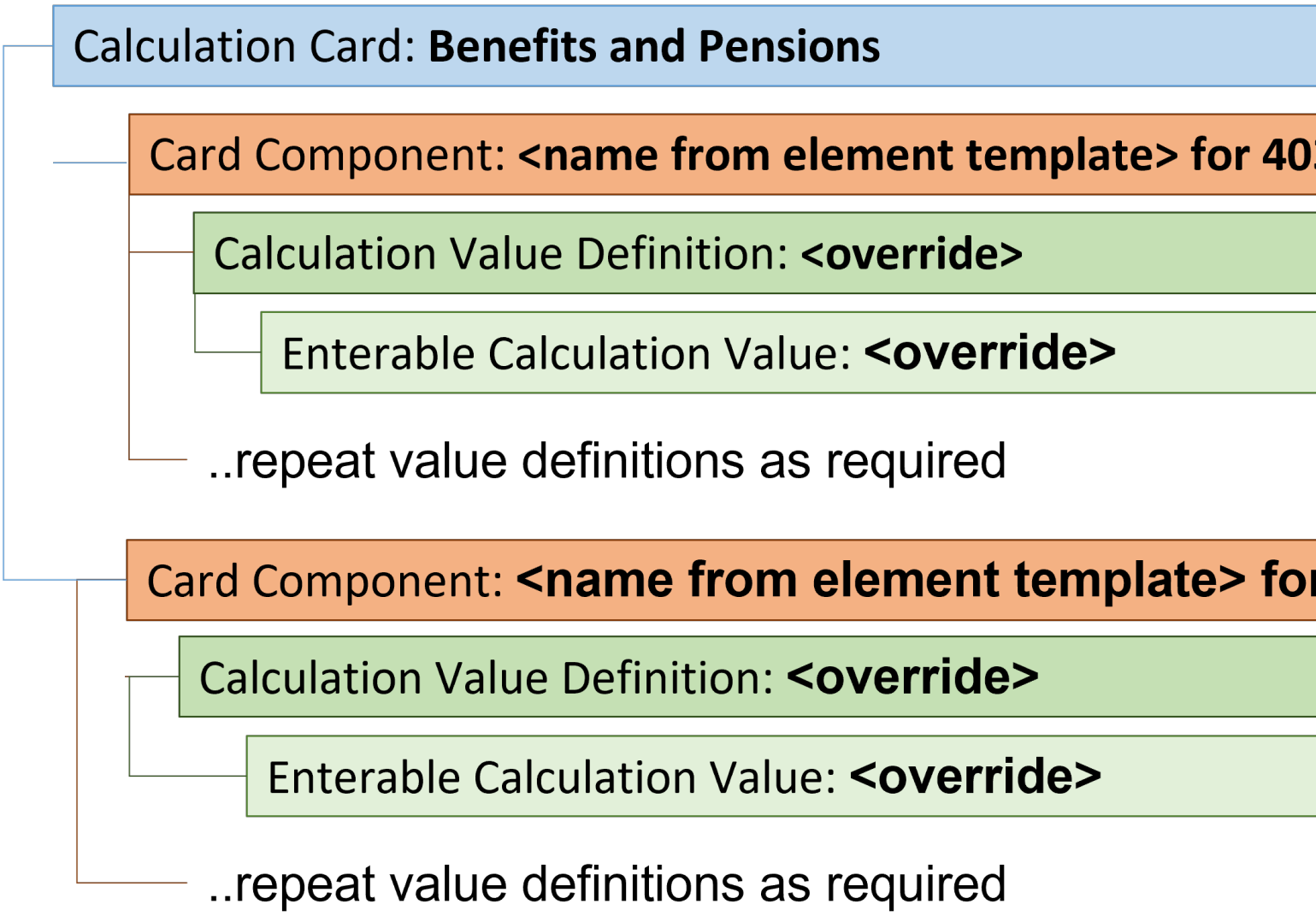
Card components on the Benefits and Pensions calculation card are used to capture 403 (b) and 457 (b) data. The Benefits and Pensions calculation card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various localization requirements. The Benefits and Pensions calculation card utilizes these Calculation Card record types for 403 (b) and 457 (b):

Calculation Card Record Types for 403 (b) and 457 (b)

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the payroll relationship for which it captures information.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections	CardComponent

Component	Functional Description	File Discriminator
	describe the card components applicable to this calculation card and the child records that are required for each card component.	
Calculation Value Definition	Allows the creation of value definitions so that overriding values can be specified on the card component. Details of the specific value definitions are provided in the following sections.	CalculationValueDefinition
Enterable Calculation Value	Used to specify an overriding value for each calculation value definition.	EnterableCalculationValue

Benefits and Pensions Calculation Card Hierarchy



The 403 (b) and 457 (b) card component names are defined by the element configuration.

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading 403 (b) Card Components
- Guidelines for Loading 457 (b) Card Components

Guidelines for Loading Benefits and Pensions 403 (b) and 457 (b) Calculation Cards

Benefits and Pensions cards are created for every employee who has elected to participate in a plan to deduct pension contributions from wages.

Even if you're updating an existing calculation card and the calculation card itself isn't being updated, include the calculation card record to group other related data supplied in the file.

Benefits and Pensions Calculation Card Attributes

The Benefits and Pensions 403 (b) and 457 (b) calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
SourceSystemId	CardSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Benefits and Pension calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify **Benefits and Pensions**.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.
PayrollRelationshipNumber(SourceSystemId)	N/A	The number that identifies the employee's payroll relationship.
EffectiveStartDate	N/A	The start date of the calculation card.

Examples of supplying these attributes are included in the card component sections.

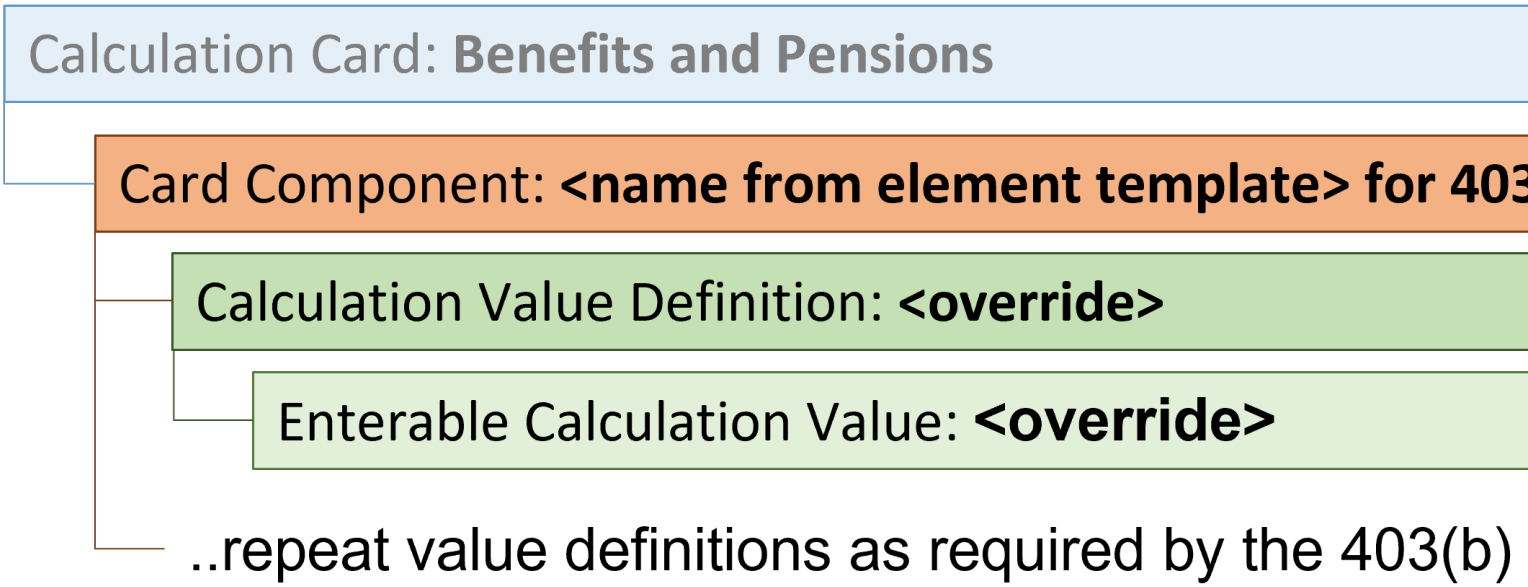
Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading 403 (b) Card Components
- Guidelines for Loading 457 (b) Card Components

Guidelines for Loading 403 (b) Card Components

The Benefits and Pensions card component is used for supporting the 403 (b) deferred compensation plans.

You can select overrides to choose a deduction type of flat amount or percentage for an employee. An override for Adjustment Deferred Compensation Limit is also available if the 15 years of service rules apply to the employee.



The 403 (b) card component is part of the Benefits and Pensions calculation card. The attributes to supply when creating a new Benefits and Pensions calculation card are described in Guidelines for Loading the Benefit and Pensions 403 (b) and 457 (b) Calculation Card topic.

The 403 (b) card component name is defined by the element template. It has multiple value definitions, which are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Card Component Attributes for 403 (b)

403 (b) Component Attributes

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the 403 (b) card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Benefits and Pensions calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the 403 (b) card component. This must be after or same as the EffectiveStartDate on the Benefits and Pensions calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Provide the name of the component as generated by the element template.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.

These attributes are supplied against the CardComponent file discriminator and must be supplied along with a CalculationCard record for the parent Benefits and Pension card. An example is provided in the Example of Loading Benefits and Pensions 403 (b) help topic.

Calculation Value Definition Attributes for 403 (b)

The Calculation Value Definition record type specifies the name of the value definition you are supplying an override value for. The Calculation Value Definition record type uses these attributes:

Calculation Value Definition Attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
SourceSystemId	CardSequence, ComponentSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent 403 (b) card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent 403 (b) card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent 403 (b) card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent **403 (b)** card component and a CalculationCard record for the owning **Benefits and Pensions** card. An example is provided in the Example of Loading Benefits and Pensions 403 (b) help topic.

Enterable Calculation Value Attributes for 403 (b)

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

Enterable Calculation Value Attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
SourceSystemId	CardSequence, ComponentSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

An example is provided in the Example of Loading Benefits and Pensions 403 (b) help topic.

Related Topics

- Guidelines for Loading Calculation Cards

Example of Loading Benefits and Pensions 403 (b)

Here's an example of how you can create the 403 (b) card component of the Benefits and Pensions card.

Source Keys are used to identify the records being loaded. The user key attributes LegislativeDataGroupName and PayrollRelationshipNumber should be used to identify the legislative data group and employee's payroll relationship.

The CalculationCard.dat file is used to bulk-load Benefit and Pension cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|EffectiveEndDate|PayrollRelationshipNumber
MERGE|CalculationCard|USER01|BP_955160008193987|PM US Sun Power|Benefits and Pensions|2022/01/01|4712/12/31|
955160008193987

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|DirCardCompDefName
MERGE|CardComponent|USER01|403b_955160008193987|PM US Sun Power|BP_955160008193987|2022/01/01|4712/12/31|
MT_403B FC

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|USER01|403b_VD1_955160008193987|PM US Sun Power|403b_955160008193987|
2022/01/01|4712/12/31|MT_403 FC|Percentage for Employee Contribution
MERGE|CalculationValueDefinition|USER01|403b_VD2_955160008193987|PM US Sun Power|403b_955160008193987|
2022/01/01|4712/12/31|MT_403 FC|Employee Catch-Up Contribution
MERGE|CalculationValueDefinition|USER01|403b_VD3_955160008193987|PM US Sun Power|403b_955160008193987|
2022/01/01|4712/12/31|MT_403 FC|Catch-Up Processing Rule

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|Value1
MERGE|EnterableCalculationValue|USER01|403b_VD1_ECV_955160008193987|PM US Sun Power|
403b_VD1_955160008193987|2022/01/01|4712/12/31|6
MERGE|EnterableCalculationValue|USER01|403b_VD2_ECV_955160008193987|PM US Sun Power|
403b_VD2_955160008193987|2022/01/01|4712/12/31|Y
MERGE|EnterableCalculationValue|USER01|403b_VD3_ECV_955160008193987|PM US Sun Power|
403b_VD3_955160008193987|2022/01/01|4712/12/31|C
```

The 403(b) CardComponent record identifies the parent Benefits and Pensions CalculationCard using the DirCardId(SourceSystemId) attribute, which has a value that matches the SourceSystemId attribute value on the CalculationCard.

The CalculationValueDefinition records identify the parent 403 (b) CardComponent using the the SourceId(SourceSystemId) attribute. The value of this attribute matches the SourceSystemId on the parent CardComponent.

The EnterableCalculationValue records identify their parent CalculationValueDefinition record using the ValueDefnId(SourceSystemId) attribute.

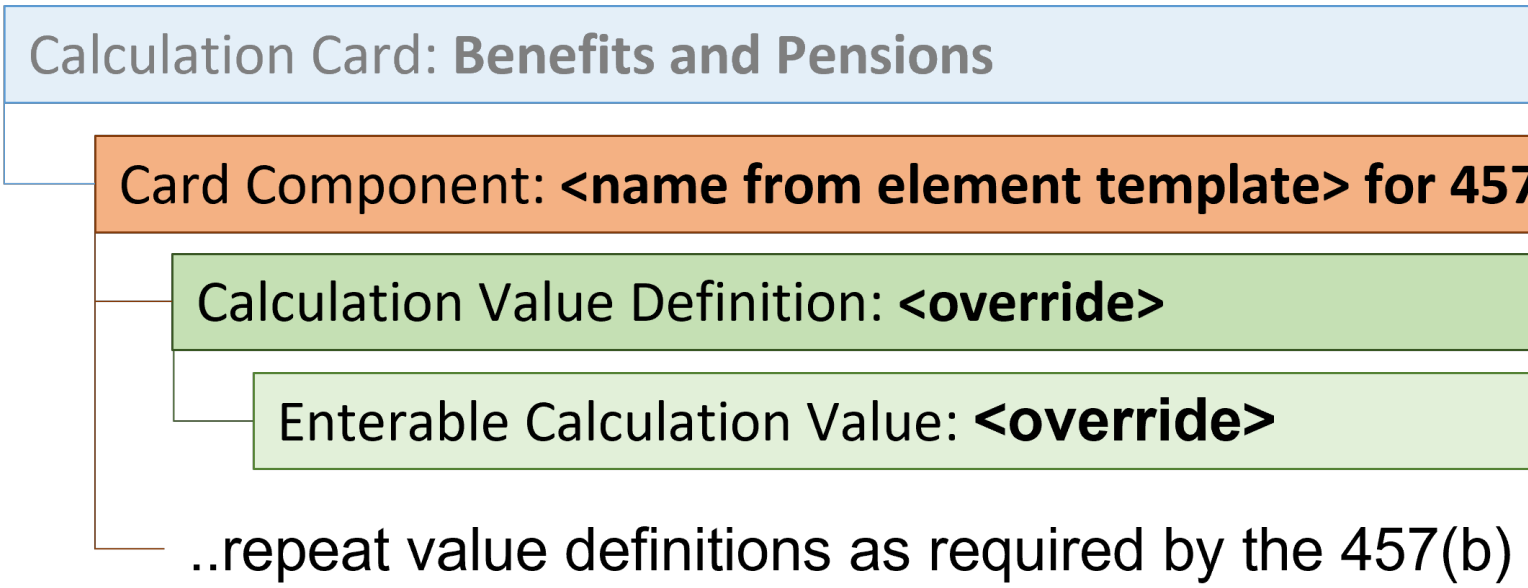
Related Topics

- Overview of Benefits and Pensions 403 (b) and 457 (b) for the US
- Guidelines for Loading Benefits and Pensions 403 (b) and 457 (b) Calculation Cards
- Guidelines for Loading 403 (b) Card Components

Guidelines for Loading 457 (b) Card Components

A 457 (b) plan is a retirement investment plan for employees of the state and municipal governments.

Although it is equivalent to a 401 (k) or a 403 (b) plan, the main structural difference is that a 457 (b) plan may allow for higher catch-up contributions.



The 457 (b) card component is part of the Benefits and Pensions calculation card. The attributes to supply when creating a new Benefits and Pensions calculation card are described in the Guidelines for Loading Benefits and Pensions 403 (b) and 457 (b) Calculation Cards help topic.

The 457 (b) card component name is defined by the element template. It has multiple value definitions, which are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Card Component Attributes for 457 (b)

The 457(b) card component uses these attributes

457 (b) Card Component Attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
SourceSystemId	CardSequence, ComponentSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the 457 (b) card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Benefits and Pensions calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The start date of the 457 (b) card component. This must be after or same as the EffectiveStartDate on the Benefits and Pensions calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Provide the name of the component as generated by the element template.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.

These attributes are supplied against the CardComponent file discriminator and must be supplied along with a CalculationCard record for the parent Benefits and Pensions card. See an example in the Example of Loading Benefits and Pensions 457 (b) help topic.

Calculation Value Definition Attributes for 457 (b)

The Calculation Value Definition record type specifies the name of the value definition you are supplying an override value for. The Calculation Value Definition record type uses these attributes

Calculation Value Definition Attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
SourceSystemId	CardSequence, ComponentSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent 457 (b) card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The start date of the parent 457 (b) card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent 457 (b) card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

Supply these attributes against the CalculationValueDefinition file discriminator and you must supply along with a CardComponent record for the parent 457 (b) card component and a CalculationCard record for the owning Benefits and Pensions card. See an example provided in the Example of Loading Benefits and Pensions 457 (b) help topic.

Enterable Calculation Value Attributes for 457 (b)

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

Enterable Calculation Value Attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
SourceSystemId	CardSequence, ComponentSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

See an example provided in the Example of Loading Benefits and Pensions 457 (b) help topic.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Benefits and Pensions 403 (b) and 457 (b) for the US
- Guidelines for Loading Benefits and Pensions 403 (b) and 457 (b) Calculation Cards

Example of Loading Benefits and Pensions 457 (b)

Here's an example of how you can create the 457 (b) card component of the Benefits and Pensions card.

Source Keys are used to identify the records that are loaded. The user key attributes LegislativeDataGroupName and PayrollRelationshipNumber should be used to identify the legislative data group and employee's payroll relationship.

The CalculationCard.dat file is used to bulk-load Benefit and Pension cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|EffectiveEndDate|PayrollRelationshipNumber
MERGE|CalculationCard|USER01|BP_955160008193987|PM US Sun Power|Benefits and Pensions|2022/01/01|4712/12/31|
955160008193987

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|DirCardCompDefName
MERGE|CardComponent|USER01|457b_955160008193987|PM US Sun Power|BP_955160008193987|2022/01/01|4712/12/31|
MT_457 LSP

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|USER01|457b_VD1_955160008193987|PM US Sun Power|457b_955160008193987|
2022/01/01|4712/12/31|MT_457 LSP|Flat Amount for Employee Contribution
MERGE|CalculationValueDefinition|USER01|457b_VD2_955160008193987|PM US Sun Power|457b_955160008193987|
2022/01/01|4712/12/31|MT_457 LSP|Partial Deduction Allowed

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|Value1
MERGE|EnterableCalculationValue|USER01|457b_VD1_ECV_955160008193987|PM US Sun Power|
457b_VD1_955160008193987|2022/01/01|4712/12/31|350
MERGE|EnterableCalculationValue|USER01|457b_VD2_ECV_955160008193987|PM US Sun Power|
457b_VD2_955160008193987|2022/01/01|4712/12/31|Y
```

The 457 (b) CardComponent record identifies the parent **Benefits and Pensions** CalculationCard using the DirCardId(SourceSystemId) attribute, which has a value that matches the SourceSystemId attribute value on the CalculationCard.

The CalculationValueDefinition records identify the parent 457 (b) CardComponent using the the SourceId(SourceSystemId) attribute. The value of this attribute matches the SourceSystemId on the parent CardComponent.

The EnterableCalculationValue records identify their parent CalculationValueDefinition record using the ValueDefnId(SourceSystemId) attribute.

Related Topics

- Overview of Benefits and Pensions 403 (b) and 457 (b) for the US
- Guidelines for Loading Benefits and Pensions 403 (b) and 457 (b) Calculation Cards
- Guidelines for Loading 457 (b) Card Components

Organization Calculation Cards

Overview of US Organization Calculation Cards

Organization calculation cards are used in the United States to store values about federal and regional tax rules required for Oracle HCM processes.

Data entered on the calculation card can be either at the Payroll Statutory Unit (PSU) or Tax Reporting Unit (TRU) calculation card level. When entered at the PSU level, the setup applies to all TRUs attached to the PSU. When entered at the TRU level, this information overrides the settings at the PSU level.

Considerations and Prerequisites

You must have previously defined the organizational structures for your business as required for HR and Payroll country-specific processes. For further information, please refer to the Implementing Payroll for the United States on My Oracle Support at the location below:

US Information Center

https://support.oracle.com/rs?type=doc&id=2063588.2

US – Payroll tab > Product Documentation > Payroll Guides > Implementing Payroll for the United States

Extract the Surrogate ID of the Tax Reporting Unit

The component sequence for the organization calculation card component is the surrogate ID of the tax reporting unit. It's an application-generated value to uniquely identify the tax reporting unit. You may use the Tax Reporting Unit BI Publisher report to extract your tax reporting units and their surrogate ID values.

> **Note:** The value may be different on different environments. For example, the value on the production pod may be different from the one on the test pod.

Generate the Vertex Geography Codes for Context Values

To load the correct context values for the US states, counties, and cities, you must provide the correct Vertex geography codes.

Calculation Card Record Types

The Calculation Rules for Tax Reporting and Payroll Statutory Unit calculation card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Calculation Rules for Tax Reporting and Payroll Statutory Unit card utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the payroll statutory unit that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Supply a component detail record for each flexfield context required by each card component.	ComponentDetail
Calculation Value Definition	Allows the creation of value definitions so that overriding values can be specified on the card component. Details of the specific value definitions are provided in the following sections.	CalculationValueDefinition
Enterable Calculation Value	For each calculation value definition an overriding value can be specified using the Enterable Calculation Value record type.	EnterableCalculationValue

Calculation Rules for Tax Reporting and Payroll Statutory Unit Hierarchy

The card hierarchy shape depends upon the level of information you're loading.

Federal State

Repeat each card component for each state you are loading data for.

Include the County and City Tax card components, if required.

Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards

Calculation Rules for Tax Reporting and Payroll Statutory Unit (PSU) card need to be created for each PSU or Tax Reporting Unit (TRU) you have defined.

Even if you are updating an existing calculation card and the calculation card itself isn't being updated, still include the calculation card record to group other related data supplied in the file.

Calculation Card Attributes

The Calculation Rules for Tax Reporting and Payroll Statutory Unit Calculation Card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, DirCardDefinitionName, LegislativeDataGroupName And PayrollStatutoryUnitName Or TaxRepUnitName	A unique identifier for the **Calculation Rules for Tax Reporting and Payroll Statutory Unit** calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The name of the card definition. Specify 'Calculation Rules for Tax Reporting and Payroll Statutory Unit'.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist. Not required when source keys are used.
PayrollStatutoryUnitName	N/A	The name of the Payroll Statutory Unit if the card is for a PSU. Don't supply for TRU level cards.
TaxRepUnitName	N/A	The name of the Tax Reporting Unit if the card is for a TRU. Don't supply for PSU level cards.
EffectiveStartDate	N/A	The start date of the calculation card.

These attributes are supplied against the CalculationCard file discriminator.

Supply the Calculation Card record along with the relevant card components as described in the following topics.

The **Calculation Rules for Tax Reporting and Payroll Statutory Unit** can havemany card components. The information that's common to them all is provided here. For guidance on the specific requirements for each card component, refer to the topics in this section.

Common Card Component Attributes

Card components commonly use these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirCardDefinitionName,	A unique identifier for the card component. For new card components supply the source

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, DirCardCompDefName And PayrollStatutoryUnitName Or TaxRepUnitName	key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, DirCardDefinitionName, LegislativeDataGroupName And PayrollStatutoryUnitName Or TaxRepUnitName	The parent Calculation Rules for Tax Reporting and Payroll Statutory Unit calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card component. This must be equal to or after the EffectiveStartDate on the Calculation Rules for Tax Reporting and Payroll Statutory Unit calculation card. If updating an existing card component, the effective start date must be original start date of the card component.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Review the guidance for card component being loaded for the value to supply.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.

These attributes are supplied against the CardComponent file discriminator.

Common Component Detail Attributes

The component detail record type allows you to upload data into flexfield segments.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield contexts used by the card component. Refer to the guidance for the card component being loaded.

You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

The component details use these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, DirCardDefinitionName, DirCardCompDefName And PayrollStatutoryUnitName Or TaxRepUnitName	A unique identifier for the component detail record. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName And PayrollStatutoryUnitName Or TaxRepUnitName	The parent card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parentcard component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component. Supply the same value as supplied to this attribute on the parent card component.
DirInformationCategory	N/A	The code for the flexfield context. Refer to the guidance for the card component for valid values.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Note: When supplying values to lookup validated flexfield segments, you can provide the lookup meaning using the '_Display' suffixed attribute name, but it is recommended that you supply the lookup code to remove potential translation issues.

Common Calculation Value Definition Attributes

The Calculation Value Definition record type specifies the value definition name for the override value.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName And PayrollStatutoryUnitName Or TaxRepUnitName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, LegislativeDataGroupName, DirCardDefinitionName, DirCardCompDefName And PayrollStatutoryUnitName Or TaxRepUnitName	The parent card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The name of the card component this value definition is for. This is used to identify the value definition and should be supplied even when source keys are used to identify the record. Supply the same value as supplied to this attribute on the parent card component.
ValueDefinitionName	N/A	The name of the value being overridden. Refer to the guidance for the card component being loaded for the list of applicable value definitions.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent **Federal Income Tax** card component and a CalculationCard record for the owning **Calculation Rules for Tax Reporting and Payroll Statutory Unit** card.

Enterable Calculation Value Attributes

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName And PayrollStatutoryUnitName Or TaxRepUnitName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName And PayrollStatutoryUnitName Or TaxRepUnitName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes.
Value1	N/A	The value for the value definition identified by the parent calculation value definition record. Unlike other calculation cards, if you supply a value definition but have no value for it supply '-999999999' to indicate a null value.

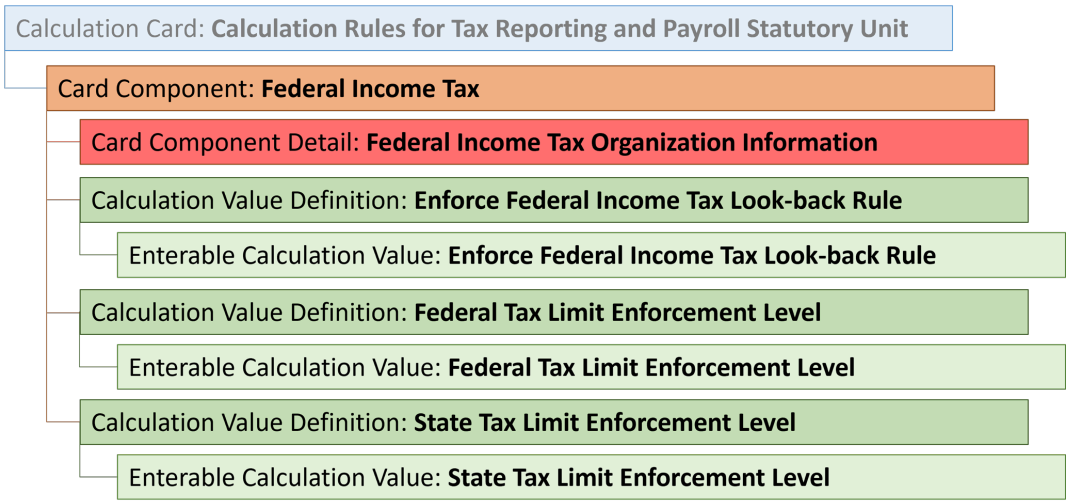
These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Guidelines for Loading Federal Income Tax Card Components

The Federal Income Tax card component is used to capture information that impacts Federal Income Tax calculations for employees.

Federal Income Tax Card Component Hierarchy

The Federal Income Tax card component has this shape:



The Federal Income Tax card component uses the Federal Income Tax Organization Information flexfield context. Data for this is loaded using the Component Detail record type. The card component also has one override, which is supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component Name

When defining the card component, specify a value of 'Federal Income Tax' for the DirCardCompDefName attribute on the Card Component, Component Detail and Calculation Value Definition records.

Component Detail Attributes

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context: Federal Income Tax Organization Information (HRX_US_ORG_FEDERAL_INCOME_TAX)

You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Federal Income Tax Value Definition

The Federal Income Tax card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Enforce Federal Income Tax Look-back Rule	For enforcing the look-back rule from the previous and current year to determine proper withholding for supplemental earnings.
Federal Tax Limit Enforcement Level	Define how the payroll process tracks limits for federal taxes.
State Tax Limit Enforcement Level	Define how the payroll process tracks limits for state taxes.

Related Topics

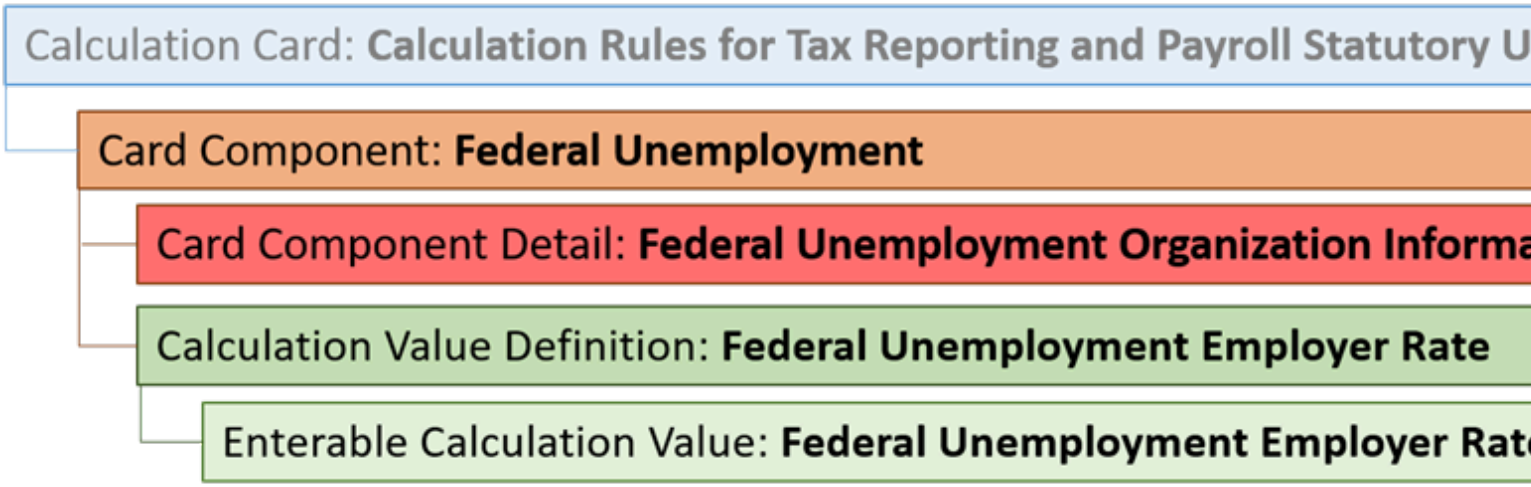
- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading Federal Rules for a Payroll Statutory Unit

Guidelines for Loading Federal Unemployment Card Components

The Federal Unemployment card component is used to capture information that impacts Federal Unemployment Tax Act (FUTA) calculations for employees.

Federal Unemployment Card Component Hierarchy

The Federal Unemployment card component has this shape:



The Federal Unemployment card component uses the Federal Unemployment Organization Information flexfield context and data for this is loaded using the Component Detail record type. It has one override, which is supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component Name

When defining the card component, specify a value of 'Federal Unemployment' for the DirCardCompDefName attribute on the Card Component, Component Detail and Calculation Value Definition records.

Component Detail Attributes

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context: Federal Unemployment Organization Information (HRX_US_ORG_FEDERAL_UNEMPLOYMENT)

You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Federal Unemployment Value Definition

The Federal Unemployment card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Federal Unemployment Employer Rate	An override of the standard FUTA employer rate at the PSU or TRU level.

Related Topics

- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading Federal Rules for a Payroll Statutory Unit

Guidelines for Loading Social Security Card Components

The Social Security card component is used to capture information that impacts Social Security calculations for employees.

Social Security Card Component Hierarchy

The Social Security card component has this shape:

The Social Security card component uses the Social Security Organization Information flexfield context and data for this is loaded using the Component Detail record type.

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component Name

When defining the card component, specify a value of 'Social Security' for the DirCardCompDefName attribute on the Card Component and Component Detail records.

Component Detail Attributes

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context: Social Security Organization Information (HRX_US_ORG_FEDERAL_SOCIAL_SECURITY)

You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Related Topics

- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading Federal Rules for a Payroll Statutory Unit

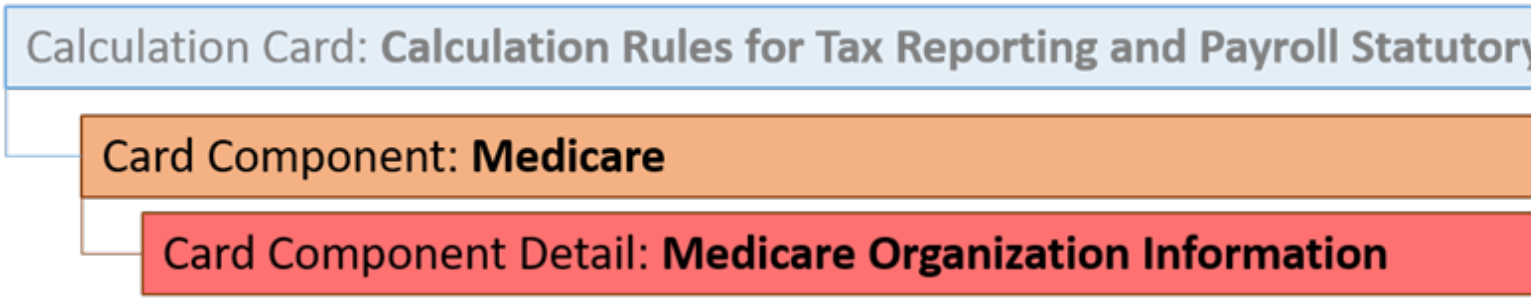
Guidelines for Loading Medicare Card Components

The Medicare card component is used to capture information that impacts Medicare calculations for employees.

Medicare Card Component Hierarchy

The Medicare card component uses the Medicare Organization Information flexfield context and data for this is loaded using the Component Detail record type.

The Medicare card component has this shape:



Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component Name

When defining the card component, specify a value of 'Medicare' for the DirCardCompDefName attribute on the Card Component and Component Detail records.

Component Detail Attributes

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context: Medicare Organization Information (ORA_HRX_US_ORG_FEDERAL_MEDICARE)

You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Related Topics

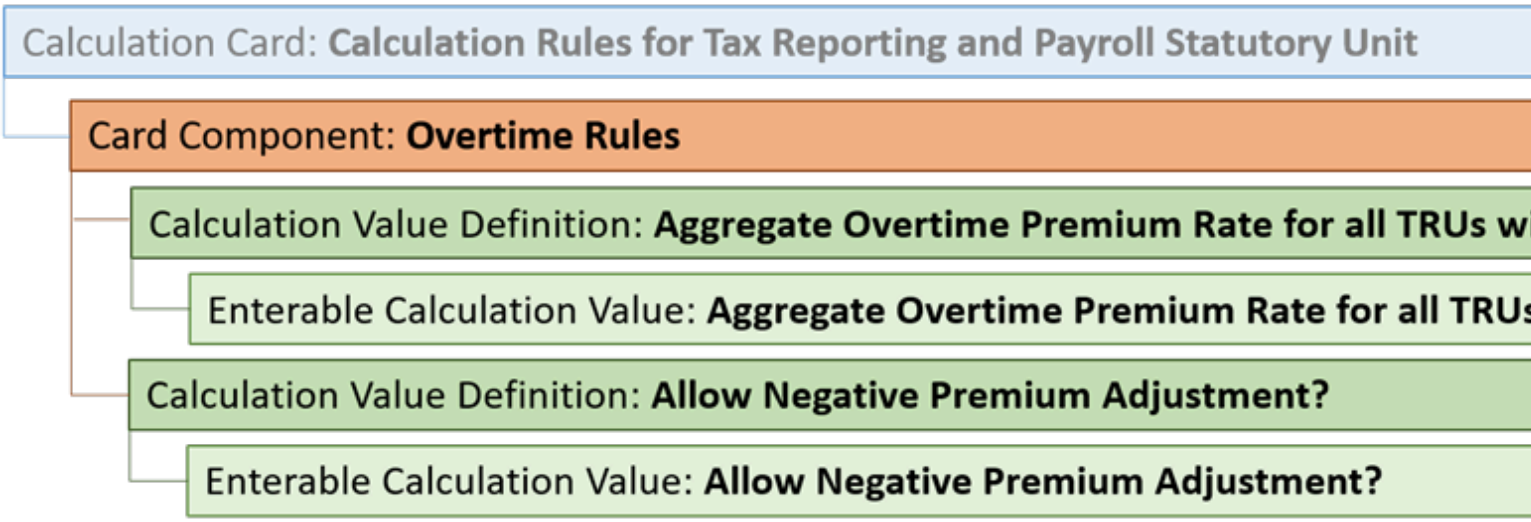
- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading Federal Rules for a Payroll Statutory Unit

Guidelines for Loading Overtime Rules Card Components

The Overtime Rules card component is used to capture information that impacts overtime calculations for employees.

Overtime Rules Card Component Hierarchy

The Overtime Rules card component has this shape:



The Overtime Rules card component uses value definitions to capture override values. Value definitions are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards topic for the attributes to supply for each of these record types.

Card Component Name

When defining the card component, specify a value of 'Overtime Rules' for the DirCardCompDefName attribute on the Card Component and Calculation Value Definition records.

Overtime Rules Value Definition

The Overtime Rules card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Aggregate Overtime Premium Rate for all TRUs within the PSU?	By setting this value to Yes, your eligible hours and earnings will aggregate across all TRUs in the same PSU. You should only enable this setting at the PSU level.
Allow Negative Premium Adjustment?	By setting this value to Yes, you are allowing negative premium adjustment calculations to be processed.

Related Topics

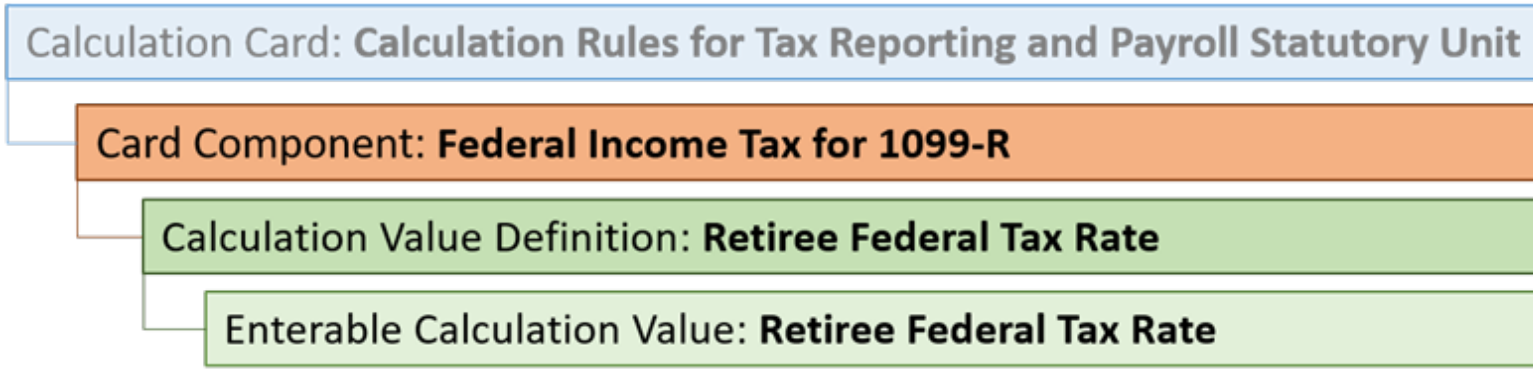
- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading Federal Rules for a Payroll Statutory Unit

Guidelines for Loading Federal Income Tax for 1099-R Card Components

The Federal Income Tax for 1099-R card component is used to capture information that impacts Federal Income Tax calculations for the 1099-R for employees.

Federal Income Tax for 1099-R Card Component Hierarchy

The Federal Income Tax for 1099-R card component has this shape:



The Federal Income Tax for 1099-R card component uses value definitions to capture override values. Value definitions are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component Name

When defining the card component, specify a value of 'Federal Income Tax for 1099-R' for the DirCardCompDefName attribute on the Card Component and Calculation Value Definition records.

Federal Income Tax for 1099-R Value Definition

The Federal Income Tax for 1099-R card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Retiree Federal Tax Rate	An override of the standard Retiree Federal Income Tax rate at the PSU or TRU level.

Related Topics

- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading Federal Rules for a Payroll Statutory Unit

Example of Loading Federal Rules for a Payroll Statutory Unit

These file lines provide an example of creating the Federal Rule card components for the Calculation Rules for Tax Reporting and Payroll Statutory Unit card for a Payroll Statutory Unit.

The following card components are created:

- Federal Income Tax
- Federal Unemployment
- Social Security
- Medicare
- Overtime Rules
- Federal Income Tax for 1099-R

The following value definitions are overridden:

Card Component	Value Definition	Override Value
Federal Unemployment	Federal Unemployment Employer Rate	10
Overtime Rules	Aggregate Overtime Premium Rate for all TRUs within the PSU?	N
Federal Income Tax for 1099-R	Retiree Federal Tax Rate	5

The CalculationCard.dat file is used to bulk-load **Calculation Rules for Tax Reporting and Payroll Statutory Unit** cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
PayrollStatutoryUnitName|EffectiveStartDate|CardSequence
MERGE|CalculationCard|VISION|USPSU_CALC_RULES|US LDG|Calculation Rules for Tax Reporting and Payroll
 Statutory Unit|US PSU|2018/01/01|1

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
PayrollStatutoryUnitName|EffectiveStartDate|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|VISION|USPSU_CALC_RULES_OVERTIME|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|Overtime
 Rules|1
MERGE|CardComponent|VISION|USPSU_CALC_RULES_1099R|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|Federal Income
 Tax for 1099-R|2
MERGE|CardComponent|VISION|USPSU_CALC_RULES_UNEMP|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|Federal
 Unemployment|3
MERGE|CardComponent|VISION|USPSU_CALC_RULES_SOC_SEC|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|Social
 Security|4
MERGE|CardComponent|VISION|USPSU_CALC_RULES_FIT|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|Federal Income
 Tax|5
MERGE|CardComponent|VISION|USPSU_CALC_RULES_MED|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|Medicare|6

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|PayrollStatutoryUnitName|EffectiveStartDate|DirCardCompDefName|
DirInformationCategory|FLEX:Deduction Developer DF|_FEDERAL_SELF_ADJUST_METHOD(Deduction
 Developer DF=HRX_US_ORG_FEDERAL_UNEMPLOYMENT)|_FEDERAL_SELF_ADJUST_METHOD(Deduction
 Developer DF=HRX_US_ORG_FEDERAL_SOCIAL_SECURITY)|_SUPPLEMENTAL_TAX_CALC_METHOD(Deduction
```

```
 Developer DF=HRX_US_ORG_FEDERAL_INCOME_TAX)|_STATE_WITHHOLDING_RULES(Deduction
 Developer DF=HRX_US_ORG_FEDERAL_INCOME_TAX)|_REGULAR_AGGREGATE_RULE(Deduction Developer
 DF=HRX_US_ORG_FEDERAL_INCOME_TAX)|_FEDERAL_SELF_ADJUST_METHOD(Deduction Developer
 DF=ORA_HRX_US_ORG_FEDERAL_MEDICARE)
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_UNEMP_DET|US LDG|USPSU_CALC_RULES_UNEMP|US PSU|2018/01/01|
Federal Unemployment|HRX_US_ORG_FEDERAL_UNEMPLOYMENT|HRX_US_ORG_FEDERAL_UNEMPLOYMENT|SELF_ADJ|||||
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_SOC_SEC_DET|US LDG|USPSU_CALC_RULES_SOC_SEC|US PSU|2018/01/01|
Social Security|HRX_US_ORG_FEDERAL_SOCIAL_SECURITY|HRX_US_ORG_FEDERAL_SOCIAL_SECURITY||SELF_ADJ||||
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_FIT_DET|US LDG|USPSU_CALC_RULES_FIT|US PSU|2018/01/01|Federal
 Income Tax|HRX_US_ORG_FEDERAL_INCOME_TAX|HRX_US_ORG_FEDERAL_INCOME_TAX|||AGGREGATION|||
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_MED_DET|US LDG|USPSU_CALC_RULES_MED|US PSU|2018/01/01|
Medicare|ORA_HRX_US_ORG_FEDERAL_MEDICARE|ORA_HRX_US_ORG_FEDERAL_MEDICARE||||||SELF_ADJ

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|PayrollStatutoryUnitName|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|USPSU_CALC_RULES_OVERTIME_VD1|US LDG|USPSU_CALC_RULES_OVERTIME|US
 PSU|2018/01/01|Overtime Rules|Aggregate Overtime Premium Rate for all TRUs within the PSU?
MERGE|CalculationValueDefinition|VISION|USPSU_CALC_RULES_1099R_VD1|US LDG|USPSU_CALC_RULES_1099R|US PSU|
2018/01/01|Federal Income Tax for 1099-R|Retiree Federal Tax Rate
MERGE|CalculationValueDefinition|VISION|USPSU_CALC_RULES_UNEMP_VD1|US LDG|USPSU_CALC_RULES_UNEMP|US PSU|
2018/01/01|Federal Unemployment|Federal Unemployment Employer Rate

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|PayrollStatutoryUnitName|EffectiveStartDate|DirCardCompDefName|
ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|VISION|USPSU_CALC_RULES_OVERTIME_VD1|US LDG|USPSU_CALC_RULES_OVERTIME_VD1|US
 PSU|2018/01/01|Overtime Rules|Aggregate Overtime Premium Rate for all TRUs within the PSU?|N
MERGE|EnterableCalculationValue|VISION|USPSU_CALC_RULES_1099R_VD1|US LDG|USPSU_CALC_RULES_1099R_VD1|US PSU|
2018/01/01|Federal Income Tax for 1099-R|Retiree Federal Tax Rate|5
MERGE|EnterableCalculationValue|VISION|USPSU_CALC_RULES_UNEMP_VD1|US LDG|USPSU_CALC_RULES_UNEMP_VD1|US PSU|
2018/01/01|Federal Unemployment|Federal Unemployment Employer Rate|10
```

Related Topics

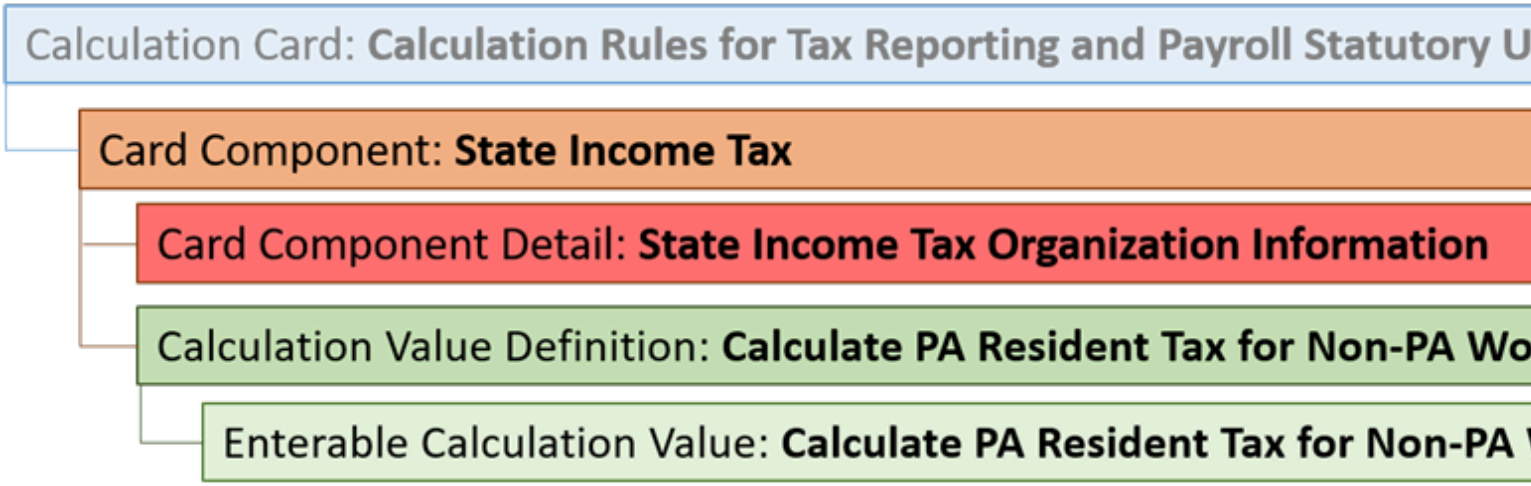
- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Guidelines for Loading Federal Income Tax Card Components
- Guidelines for Loading Federal Unemployment Card Components
- Guidelines for Loading Social Security Card Components
- Guidelines for Loading Medicare Card Components
- Guidelines for Loading Overtime Rules Card Components

Guidelines for Loading State Income Tax Card Components

The State Income Tax card component is used to capture information that impacts State Income Tax calculations for employees.

State Income Tax Card Component Hierarchy

The State Income Tax card component has this shape:



The State Income Tax card component uses the State Income Tax Organization Information flexfield context and data for this is loaded using the Component Detail record type. For the state of Pennsylvania you can also provide an override for the Calculate PA Resident Tax for Non-PA Work Location value definition, this is loaded using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component

When defining the card component, specify a value of 'State Income Tax' for the DirCardCompDefName attribute on the Card Component and Component Detail records.

In addition to the common card component attributes, you are required to supply the state information:

HCM Data Loader Attribute	Functional Description
Context1	The geocode number that identifies the US state. Refer to Cloud Customer Connect topic Report: US State, County and City geographical codes

Component Detail Attributes

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context:

- State Income Tax Organization Information (HRX_US_ORG_STATE_INCOME_TAX)

You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

State Income Tax Value Definition

The State Income Tax card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Calculate PA Resident Tax for Non-PA Work Location	To withhold local taxes for Pennsylvania residents who work outside of Pennsylvania as a courtesy, set this value to Y

Related Topics

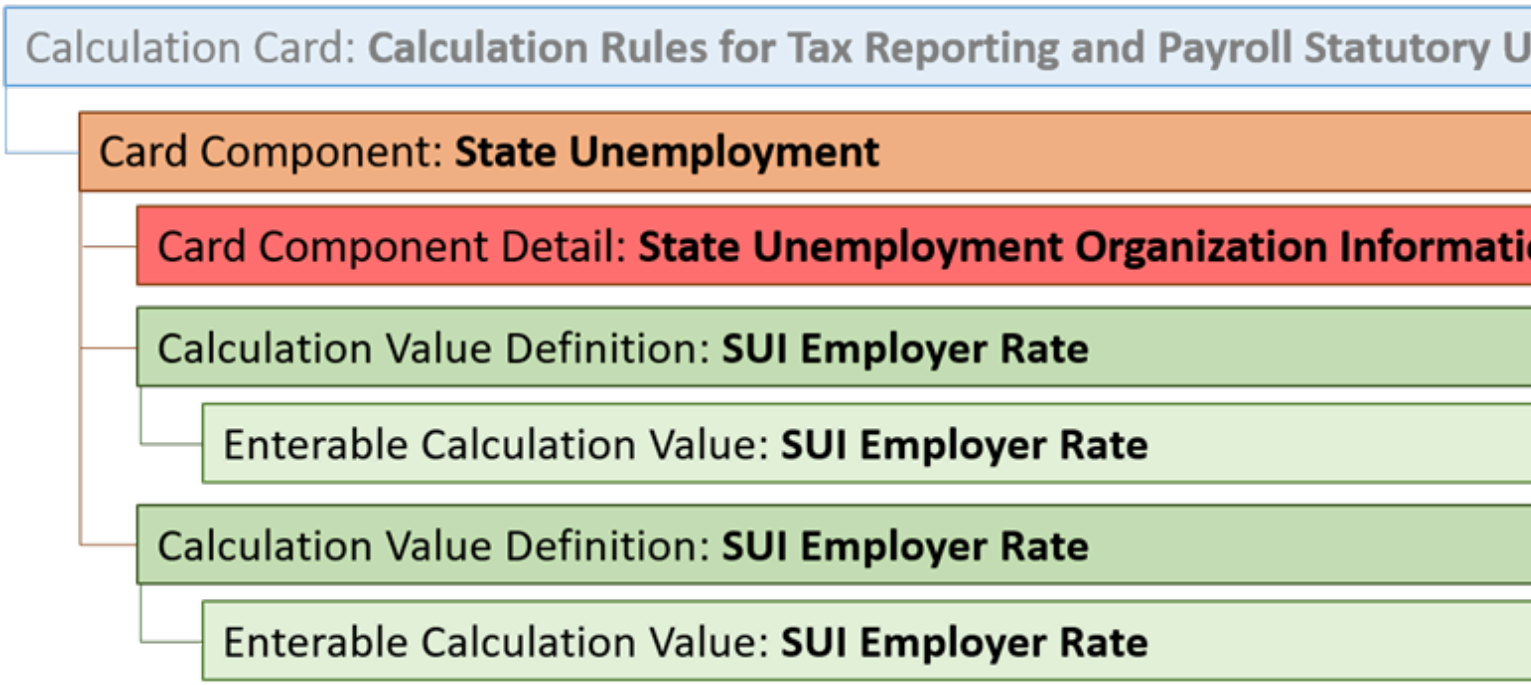
- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading State Rules for a Payroll Statutory Unit

Guidelines for Loading State Unemployment Card Components

The State Unemployment card component is used to capture information that impacts State Unemployment Tax calculations for employees.

State Unemployment Card Component Hierarchy

The State Unemployment card component has this shape:



The State Unemployment card component uses the State Unemployment Organization Information flexfield context and data for this is loaded using the Component Detail record type. You provide overrides for the value definitions using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component

When defining the card component, specify a value of '**State Unemployment**' for the DirCardCompDefName attribute on the Card Component, Component Detail and Calculation Value Definition records.

In addition to the common card component attributes, you are required to supply the state information:

HCM Data Loader Attribute	Functional Description
Context1	The geocode number that identifies the US state. Refer to Cloud Customer Connect topic Report: US State, County and City geographical codes.

Component Detail Attributes

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context: State Unemployment Organization Information (HRX_US_ORG_STATE_UNEMPLOYMENT)

You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

State Unemployment Value Definition

The State Unemployment card component uses value definitions to supply override values.

Value Definition Name	Functional Description
SUI Employer Rate	The SUI employer rate used to calculate the employer's state unemployment liability. You can define this rate at the PSU or TRU level.
SUI Employee Rate	An override to the standard SUI employee rate. This value definition is applicable to AK, NJ and PA only.

When supplying Calculation Value Definition records, in addition to the common attributes, you are required to supply the state information by including this additional attribute:

HCM Data Loader Attribute	Functional Description
Context1	Supply the same value as the Context1 attribute on the parent CardComponent record.

Related Topics

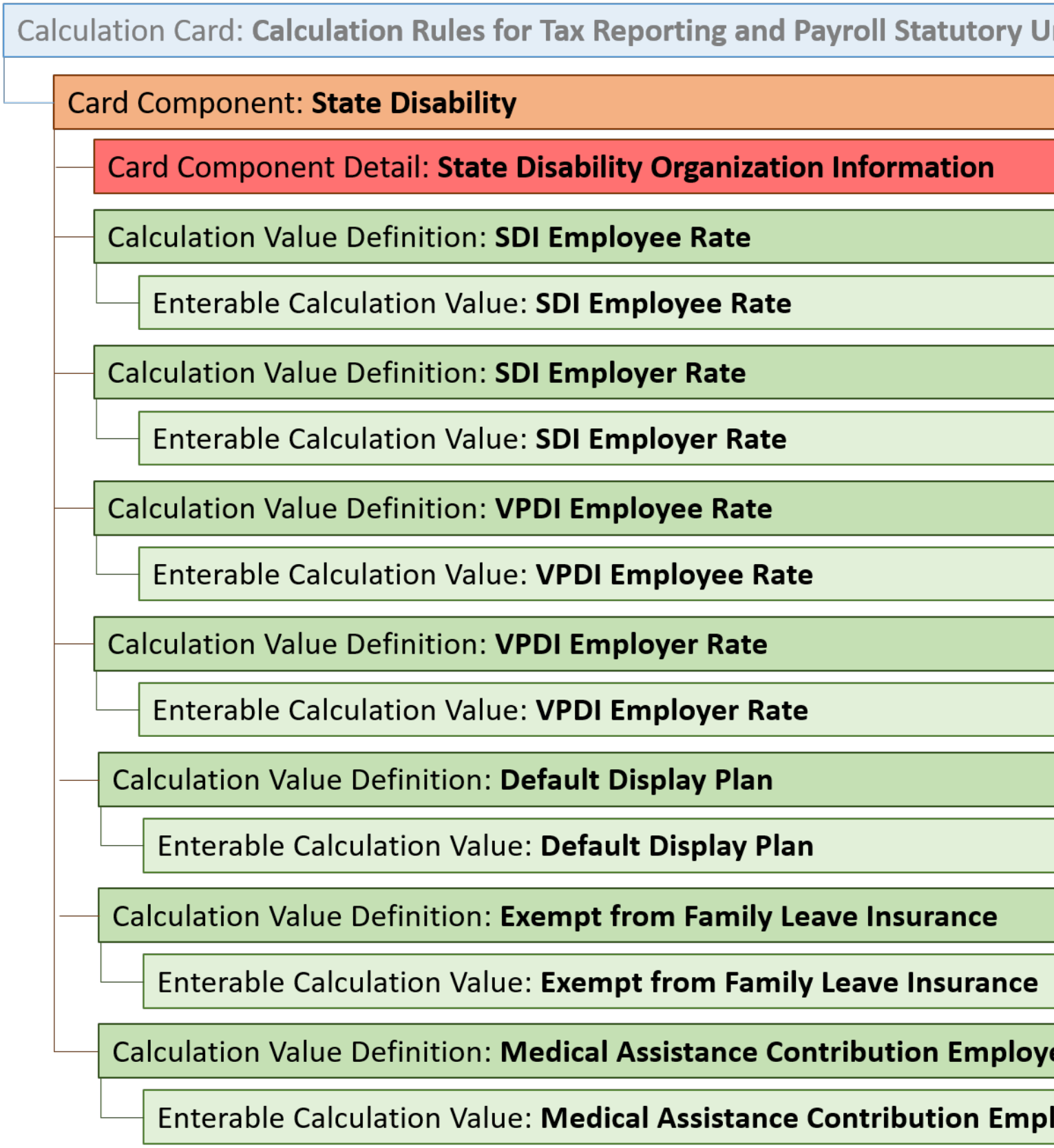
- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading State Rules for a Payroll Statutory Unit

Guidelines for Loading State Disability Card Components

The State Disability card component is used to capture information that impacts State Disability Insurance (SDI) calculations for employees.

State Disability Card Component Hierarchy

The State Disability card component has this shape:


Calculation Card: Calculation Rules for Tax Reporting and Payroll Statutory U
Card Component: State Disability
Card Component Detail: State Disability Organization Information
Calculation Value Definition: SDI Employee Rate
Enterable Calculation Value: SDI Employee Rate
Calculation Value Definition: SDI Employer Rate
Enterable Calculation Value: SDI Employer Rate
Calculation Value Definition: VPDI Employee Rate
Enterable Calculation Value: VPDI Employee Rate
Calculation Value Definition: VPDI Employer Rate
Enterable Calculation Value: VPDI Employer Rate
Calculation Value Definition: Default Display Plan
Enterable Calculation Value: Default Display Plan
Calculation Value Definition: Exempt from Family Leave Insurance
Enterable Calculation Value: Exempt from Family Leave Insurance
Calculation Value Definition: Medical Assistance Contribution Employe
Enterable Calculation Value: Medical Assistance Contribution Empl

The State Disability card component uses the State Disability Organization Information flexfield context and data for this is loaded using the Component Detail record type. You provide overrides for the value definitions using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** help topic for the attributes to supply for each of these record types.

Card Component

When defining the card component, specify a value of '**State Disability**' for the DirCardCompDefName attribute on the Card Component, Component Detail and Calculation Value Definition records.

In addition to the common card component attributes, you are required to supply the state information:

HCM Data Loader Attribute	Functional Description
Context1	The geocode number that identifies the US state. Refer to Cloud Customer Connect topic Report: US State, County and City geographical codes.

State Disability Value Definition

The State Disability card component uses value definitions to supply override values.

Value Definition Name	Functional Description
SDI Employee Rate	Used for the SDI employee rate. Applicable to states CA, HI, NJ, NY and RI only.
SDI Employer Rate	Used for the SDI employer rate. Applicable to NJ only.
VPDI Employee Rate	Used for the VPDI employee rate. Applicable to CA only.
VPDI Employer Rate	Used for the VPDI employer rate. Applicable to CA only.
Default Disability Plan	Used to override the state disability tax deductions. Applicable to CA only.
Exempt from Family Leave Insurance	Exempt the employees from Family Leave Insurance. For NY, set this value to N to withhold. Otherwise, set it to Y to not withhold. For NJ, set the value to Y if you don't want to withhold.
Medical Assistance Contribution Employer Rate	Medical Assistance Contribution Employer Rate. Applicable to MA only.

Related Topics

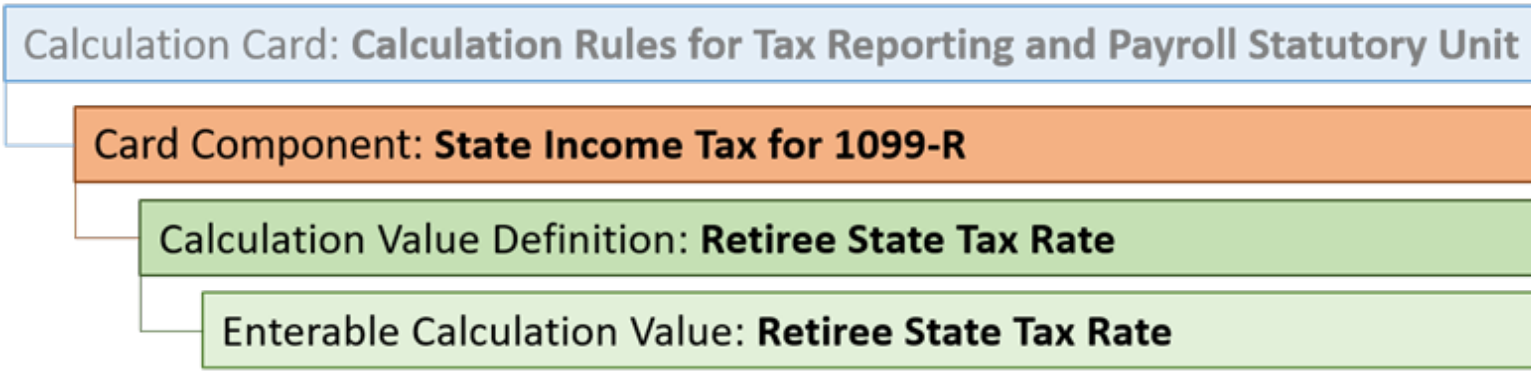
- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading State Rules for a Payroll Statutory Unit

Guidelines for Loading State Income Tax for 1099-R Card Components

The State Income Tax for 1099-R card component is used to capture information that impacts State Income Tax calculations for the 1099-R for employees.

State Income Tax for 1099-R Card Component Hierarchy

The State Income Tax for 1099-R card component has this shape:



The State Income Tax for 1099-R card component uses value definitions to capture override values. Value definitions are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component

When defining the card component, specify a value of 'State Income Tax for 1099-R' for the DirCardCompDefName attribute on the Card Component and Calculation Value Definition records.

In addition to the common card component attributes, you're required to supply the state information by including this additional attribute:

HCM Data Loader Attribute	Functional Description
Context1	The geocode number that identifies the US state. Refer to Cloud Customer Connect topic Report: US State, County and City geographical codes.

State Income Tax for 1099-R Value Definition

The State Income Tax for 1099-R' card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Retiree State Tax Rate	An override of the standard Retiree State Income Tax rate at the PSU or TRU level.

When supplying Calculation Value Definition records, in addition to the common attributes, you're required to supply the state information by including this additional attributes:

HCM Data Loader Attribute	Functional Description
Context1	Supply the same value as the Context1 attribute on the parent CardComponent record.

Related Topics

- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading State Rules for a Payroll Statutory Unit

Example of Loading State Rules for a Payroll Statutory Unit

Here's an example where you can use HCM Data Loader to load state rules for a payroll statutory unit.

These file lines provide an example of creating the State rule card components for the **Calculation Rules for Tax Reporting and Payroll Statutory Unit** card for a Payroll Statutory Unit. This example is for the state of Colorado.

The CalculationCard.dat file is used to bulk-load **Calculation Rules for Tax Reporting and Payroll Statutory Unit** cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
PayrollStatutoryUnitName|EffectiveStartDate|CardSequence
MERGE|CalculationCard|VISION|USPSU_CALC_RULES|US LDG|Calculation Rules for Tax Reporting and Payroll
 Statutory Unit|US PSU|2018/01/01|1

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
PayrollStatutoryUnitName|EffectiveStartDate|Context1|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|VISION|USPSU_CALC_RULES_6_UNEMP|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|6|State
 Unemployment|1
MERGE|CardComponent|VISION|USPSU_CALC_RULES_6_SIT|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|6|State Income
 Tax|2
MERGE|CardComponent|VISION|USPSU_CALC_RULES_6_DSBLTY|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|6|State
 Disability|3

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|PayrollStatutoryUnitName|EffectiveStartDate|DirInformationCategory|
FLEX:Deduction Developer DF|_STATE_SELF_ADJUST_METHOD(Deduction Developer DF=HRX_US_ORG_STATE_UNEMPLOYMENT)|
_SUPPLEMENTAL_TAX_CALC_METHOD(Deduction Developer DF=HRX_US_ORG_STATE_INCOME_TAX)|
_ORG_STATE_RESIDENT_WAGE_ACCUM(Deduction Developer DF=HRX_US_ORG_STATE_INCOME_TAX)|
_COUNTY_WITHHOLDING_RULES(Deduction Developer DF=HRX_US_ORG_STATE_INCOME_TAX)|
_CITY_WITHHOLDING_RULES(Deduction Developer DF=HRX_US_ORG_STATE_INCOME_TAX)|
_STATE_SELF_ADJUST_METHOD(Deduction Developer DF=HRX_US_ORG_STATE_DISABILITY)
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_6_UNEMP_DET|US LDG|USPSU_CALC_RULES_6_UNEMP|US PSU|2018/01/01|
HRX_US_ORG_STATE_UNEMPLOYMENT|HRX_US_ORG_STATE_UNEMPLOYMENT|SELF_ADJ|||||
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_6_SIT_DET|US LDG|USPSU_CALC_RULES_6_SIT|US PSU|2018/01/01|
HRX_US_ORG_STATE_INCOME_TAX|HRX_US_ORG_STATE_INCOME_TAX||AGGREGATION|DEFAULT|ALL_COUNTIES|ALL_CITIES|
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_6_DSBLTY_DET|US LDG|USPSU_CALC_RULES_6_DSBLTY|US PSU|
2018/01/01|HRX_US_ORG_STATE_DISABILITY|HRX_US_ORG_STATE_DISABILITY||||||SELF_ADJ

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|PayrollStatutoryUnitName|EffectiveStartDate|Context1|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|USPSU_CALC_RULES_6_UNEMP_VD1|US LDG|USPSU_CALC_RULES_6_UNEMP|US PSU|
2018/01/01|6|SUI Employer Rate
```

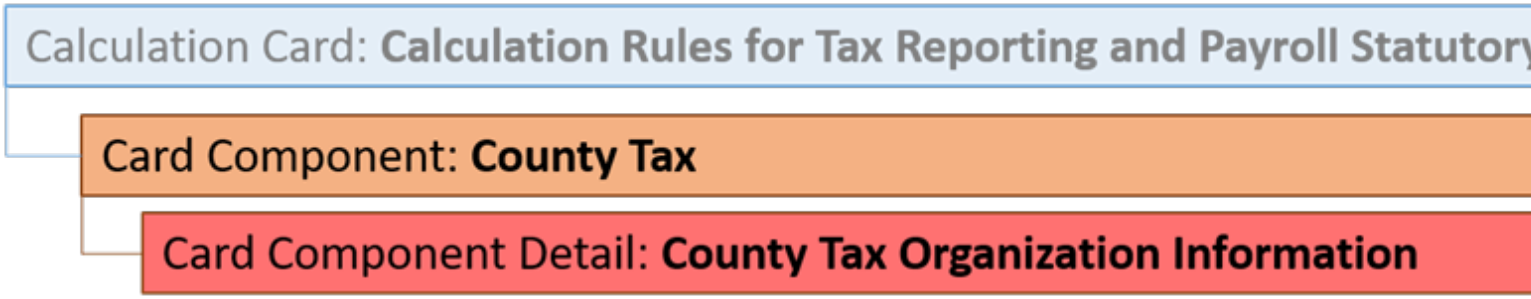
```
METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|PayrollStatutoryUnitName|EffectiveStartDate|Context1|ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|VISION|USPSU_CALC_RULES_6_UNEMP_VD1|US LDG|USPSU_CALC_RULES_6_UNEMP_VD1|US
 PSU|2018/01/01|6|SUI Employer Rate|5
```

Guidelines for Loading County Tax Card Components

The County Tax card component is used to capture information that impacts county tax calculations for employees. This regional component is defined at the state and county level.

County Tax Card Component Hierarchy

The County Tax card component has this shape:



The County Tax card component uses the County Tax Organization Information flexfield context and data for this is loaded using the Component Detail record type.

Refer to Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards topic for the attributes to supply for each of these record types.

Card Component

When defining the card component, specify a value of 'County Tax' for the DirCardCompDefName attribute on the Card Component and Component Detail records.

In addition to the common card component attributes, you are required to supply the state and county identifiers:

HCM Data Loader Attribute	Functional Description
Context1	The geocode number that identifies the US state.
Context2	The geocode number that identifies the US county.

Component Detail Attributes

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context: County Tax Organization Information (HRX_US_ORG_COUNTY_TAX)

You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Related Topics

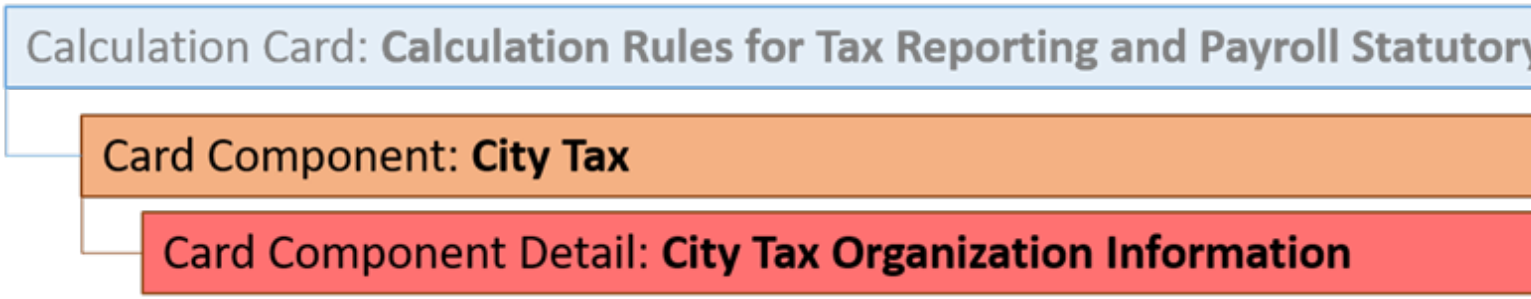
- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading State Rules for a Payroll Statutory Unit

Guidelines for Loading City Tax Card Components

The City Tax card component is used to capture information that impacts city tax calculations for employees. This regional component is defined at the state, county, and city level.

City Tax Card Component Hierarchy

The City Tax card component has this shape:



The City Tax card component uses the City Tax Organization Information flexfield context and data for this is loaded using the Component Detail record type.

Refer to Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards topic for the attributes to supply for each of these record types.

Card Component

When defining the card component, specify a value of 'City Tax' for the DirCardCompDefName attribute on the Card Component and Component Detail records.

In addition to the common card component attributes, you are required to supply the state, county and city identifiers:

HCM Data Loader Attribute	Functional Description
Context1	The geocode number that identifies the US state.
Context2	The geocode number that identifies the US county.
Context3	The geocode number that identifies the US city.

Component Detail Attributes

In addition to the common attributes for the component details record include the necessary flexfield segment attribute values for the flexfield context: City Tax Organization Information (HRX_US_ORG_CITY_TAX)

You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Related Topics

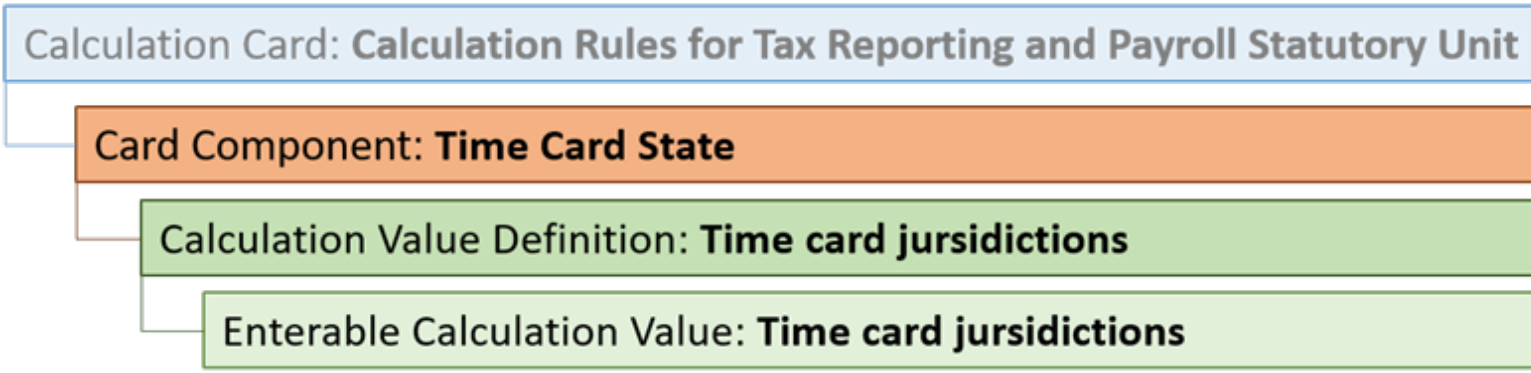
- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading State Rules for a Payroll Statutory Unit

Guidelines for Loading Time Card State Card Components

The Time Card State card component is used to capture information that defines the jurisdictions on the time card that impacts tax calculations for employees.

Time Card State Card Component Hierarchy

The Time Card State card component has this shape:



The Time Card State card component uses value definitions to capture override values. Value definitions are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component

When defining the card component, specify a value of 'Time Card State' for the DirCardCompDefName attribute on the Card Component and Calculation Value Definition records.

In addition to the common card component attributes, you are required to supply the state information:

HCM Data Loader Attribute	Functional Description
Context1	The geocode number that identifies the US state. Refer to Cloud Customer Connect topic Report: US State, County and City geographical codes.

Time Card State Value Definition

The Time Card State card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Time card jurisdictions	Select the appropriate behavior for this time card jurisdiction. Options are: Exclude state, Include state, Include all counties, Include all counties and cities. When you select Include all counties or all counties and cities, you are selecting all counties and / or cities for only that state.

Related Topics

- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading State Rules for a Payroll Statutory Unit

Guidelines for Loading Time Card County Card Components

The Time Card County card component is used to capture information that defines the jurisdictions on the time card that impacts tax calculations for employees. This regional component is defined at the county level.

Time Card County Card Component Hierarchy

The Time Card County card component uses value definitions to capture override values. Value definitions are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component

When defining the card component, specify a value of '**Time Card County**' for the DirCardCompDefName attribute on the Card Component and Calculation Value Definition records.

In addition to the common card component attributes, you are required to supply the state information:

HCM Data Loader Attribute	Functional Description
Context1	The geocode number that identifies the US state.
Context2	The geocode number that identifies the US county.

Time Card County Value Definition

The Time Card County card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Time card jurisdictions	Select the appropriate behavior for this time card jurisdiction. Options are: Exclude county, Include county, Include all cities. When you select Include all cities, you are selecting all cities for only that county.

Related Topics

- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading State Rules for a Payroll Statutory Unit

Guidelines for Loading Time Card City Card Components

The Time Card City card component is used to capture information that defines the jurisdictions on the time card that impacts tax calculations for employees.

This regional component is defined at the state, county, and city level.

The **Time Card City** card component uses value definitions to capture override values. Value definitions are supplied using the Calculation Value Definition and Enterable Calculation Value record types

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component

When defining the card component, specify a value of '**Time Card City'** for the DirCardCompDefName attribute on the Card Component and Calculation Value Definition records.

In addition to the common card component attributes, you're required to supply the state information:

HCM Data Loader Attribute	Functional Description
Context1	The geocode number that identifies the US state.
Context2	The geocode number that identifies the US county.
Context3	The geocode number that identifies the US city.

Time Card County Value Definition

The Time Card County card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Time card jurisdictions	Select the appropriate behavior for this time card jurisdiction. Options are: Exclude city, Include city.

Related Topics

- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading State Rules for a Payroll Statutory Unit

Guidelines for Loading Pension Plan Card Components

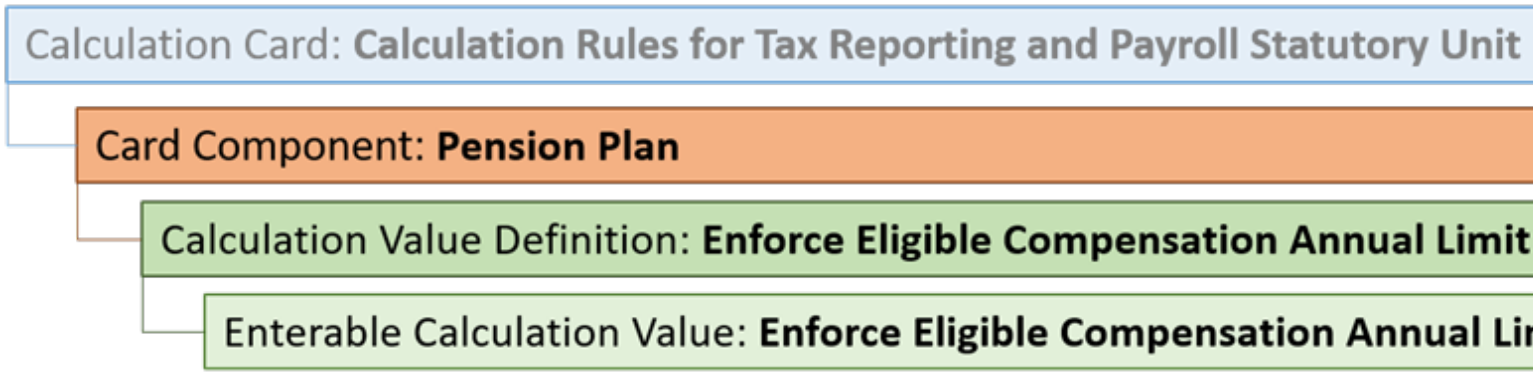
The Pension Plan card component is used to capture information that captures details for pension plans.

This component captures details for the following kinds of Pension Types that are defined at the time the Pension Plan component is created:

- 401(k)
- 403(b)
- 457

Pension Plan Card Component Hierarchy

The Pension Plan card component has this shape:



The Pension Plan card component uses value definitions to capture override values. Value definitions are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to **Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards** topic for the attributes to supply for each of these record types.

Card Component

When defining the card component, specify a value of 'Pension Plan' for the DirCardCompDefName attribute on the Card Component and Calculation Value Definition records.

In addition to the common card component attributes, you are required to supply the state information:

HCM Data Loader Attribute	Functional Description
Context1	The Pension Type of the Pension Plan card component. Valid values are 401(k), 403(b), and 457.

Pension Plan Value Definition

The Pension Plan card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Enforce Eligible Compensation Annual Limit Check	Specify whether or not to enforce the compensation annual limit. Enter Y or N.

Related Topics

- Guidelines for Loading Calculation Rules for Tax Reporting and Payroll Statutory Unit Cards
- Example of Loading State Rules for a Payroll Statutory Unit

Example of Loading Local Rules for a Payroll Statutory Unit

This example file creates the Calculation Rules for Tax Reporting and Payroll Statutory Unit calculation card for the PSU and defines the state and local rules for the state of California.

The following card components are created:

- State Income Tax
- State Unemployment
- State Disability
- County Tax
- City Tax

The following value definitions are overridden:

Card Component	Value Definition	Override Value
State Disability	SDI Employer Rate	6.5
State Unemployment	SDI Employer Rate	4.5

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
PayrollStatutoryUnitName|EffectiveStartDate|CardSequence
MERGE|CalculationCard|VISION|USPSU_CALC_RULES|US LDG|Calculation Rules for Tax Reporting and Payroll
 Statutory Unit|US PSU|2018/01/01|1

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
PayrollStatutoryUnitName|EffectiveStartDate|DirCardCompDefName|ComponentSequence|Context1|Context2|Context3
MERGE|CardComponent|VISION|USPSU_CALC_RULES_5_UNEMP|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|State
 Unemployment|1|5||
MERGE|CardComponent|VISION|USPSU_CALC_RULES_5_SIT|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|State Income
 Tax|2|5||
MERGE|CardComponent|VISION|USPSU_CALC_RULES_5_DSBLTY|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|State
 Disability|3|5||
MERGE|CardComponent|VISION|USPSU_CALC_RULES_5_75_TAX|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|County Tax|4|
5|75|
MERGE|CardComponent|VISION|USPSU_CALC_RULES_5_75_8922_TAX|US LDG|USPSU_CALC_RULES|US PSU|2018/01/01|City
 Tax|5|5|75|8922

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|PayrollStatutoryUnitName|EffectiveStartDate|DirInformationCategory|
FLEX:Deduction Developer DF|_STATE_SELF_ADJUST_METHOD(Deduction Developer DF=HRX_US_ORG_STATE_UNEMPLOYMENT)|
_SUPPLEMENTAL_TAX_CALC_METHOD(Deduction Developer DF=HRX_US_ORG_STATE_INCOME_TAX)|
_ORG_STATE_RESIDENT_WAGE_ACCUM(Deduction Developer DF=HRX_US_ORG_STATE_INCOME_TAX)|
_COUNTY_WITHHOLDING_RULES(Deduction Developer DF=HRX_US_ORG_STATE_INCOME_TAX)|
_CITY_WITHHOLDING_RULES(Deduction Developer DF=HRX_US_ORG_STATE_INCOME_TAX)|
_STATE_SELF_ADJUST_METHOD(Deduction Developer DF=HRX_US_ORG_STATE_DISABILITY)|_THRESHOLD_HOURS(Deduction
 Developer DF=HRX_US_ORG_COUNTY_TAX)|_ORG_COUNTY_RESIDENT_WAGE_ACCUM(Deduction Developer
 DF=HRX_US_ORG_COUNTY_TAX)|_THRESHOLD_HOURS(Deduction Developer DF=HRX_US_ORG_CITY_TAX)|
_ORG_CITY_RESIDENT_WAGE_ACCUM(Deduction Developer DF=HRX_US_ORG_CITY_TAX)
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_5_UNEMP_DET|US LDG|USPSU_CALC_RULES_5_UNEMP|US PSU|2018/01/01|
HRX_US_ORG_STATE_UNEMPLOYMENT|HRX_US_ORG_STATE_UNEMPLOYMENT|SELF_ADJ|||||||||
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_5_SIT_DET|US LDG|USPSU_CALC_RULES_5_SIT|US PSU|2018/01/01|
HRX_US_ORG_STATE_INCOME_TAX|HRX_US_ORG_STATE_INCOME_TAX||AGGREGATION|DEFAULT|ALL_COUNTIES|ALL_CITIES|||||
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_5_DSBLTY_DET|US LDG|USPSU_CALC_RULES_5_DSBLTY|US PSU|
2018/01/01|HRX_US_ORG_STATE_DISABILITY|HRX_US_ORG_STATE_DISABILITY||||||SELF_ADJ||||
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_5_75_TAX_DET|US LDG|USPSU_CALC_RULES_5_75_TAX|US PSU|
2018/01/01|HRX_US_ORG_COUNTY_TAX|HRX_US_ORG_COUNTY_TAX|||||||15|DEFAULT||
MERGE|ComponentDetail|VISION|USPSU_CALC_RULES_5_75_8922_TAX_DET|US LDG|USPSU_CALC_RULES_5_75_8922_TAX|US
 PSU|2018/01/01|HRX_US_ORG_CITY_TAX|HRX_US_ORG_CITY_TAX|||||||||15|DEFAULT

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|PayrollStatutoryUnitName|EffectiveStartDate|Context1|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|USPSU_CALC_RULES_5_DSBLTY_VD1|US LDG|USPSU_CALC_RULES_5_DSBLTY|US
 PSU|2018/01/01|5|SDI Employee Rate
MERGE|CalculationValueDefinition|VISION|USPSU_CALC_RULES_5_UNEMP_VD1|US LDG|USPSU_CALC_RULES_5_UNEMP|US PSU|
2018/01/01|5|SUI Employer Rate

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|PayrollStatutoryUnitName|EffectiveStartDate|Context1|ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|VISION|USPSU_CALC_RULES_5_DSBLTY_VD1|US LDG|USPSU_CALC_RULES_5_DSBLTY_VD1|US
 PSU|2018/01/01|5|SDI Employee Rate|6.5
MERGE|EnterableCalculationValue|VISION|USPSU_CALC_RULES_5_UNEMP_VD1|US LDG|USPSU_CALC_RULES_5_UNEMP_VD1|US
 PSU|2018/01/01|5|SUI Employer Rate|4.5
```

Employee Earnings Distribution Overrides

Overview of Employee Earnings Distribution Overrides Card

The Employee Earnings Distribution Overrides calculation card is used to derive the percentage of earnings to be distributed over different jurisdictions.

Data entered on the calculation card can be entered at three geography levels:

- State
- County
- City

This card must be created for each payroll relationship the employee is in. Details are captured on the Regional component. For example, you may create a Regional component to distribute 50% of an employee's earnings to a different state.

Extracting the Surrogate ID of the Tax Reporting Unit

The component sequence for the organization calculation card component is the surrogate ID of the tax reporting unit. It is an application-generated value to uniquely identify the tax reporting unit. You may use the Tax Reporting Unit BI Publisher report to extract your tax reporting units and their surrogate ID values.

> **Note:** The value may be different on different environments. For example, the value on the production environment may be different from the one on the test environment.

Employee Earnings Distribution Overrides Record Types

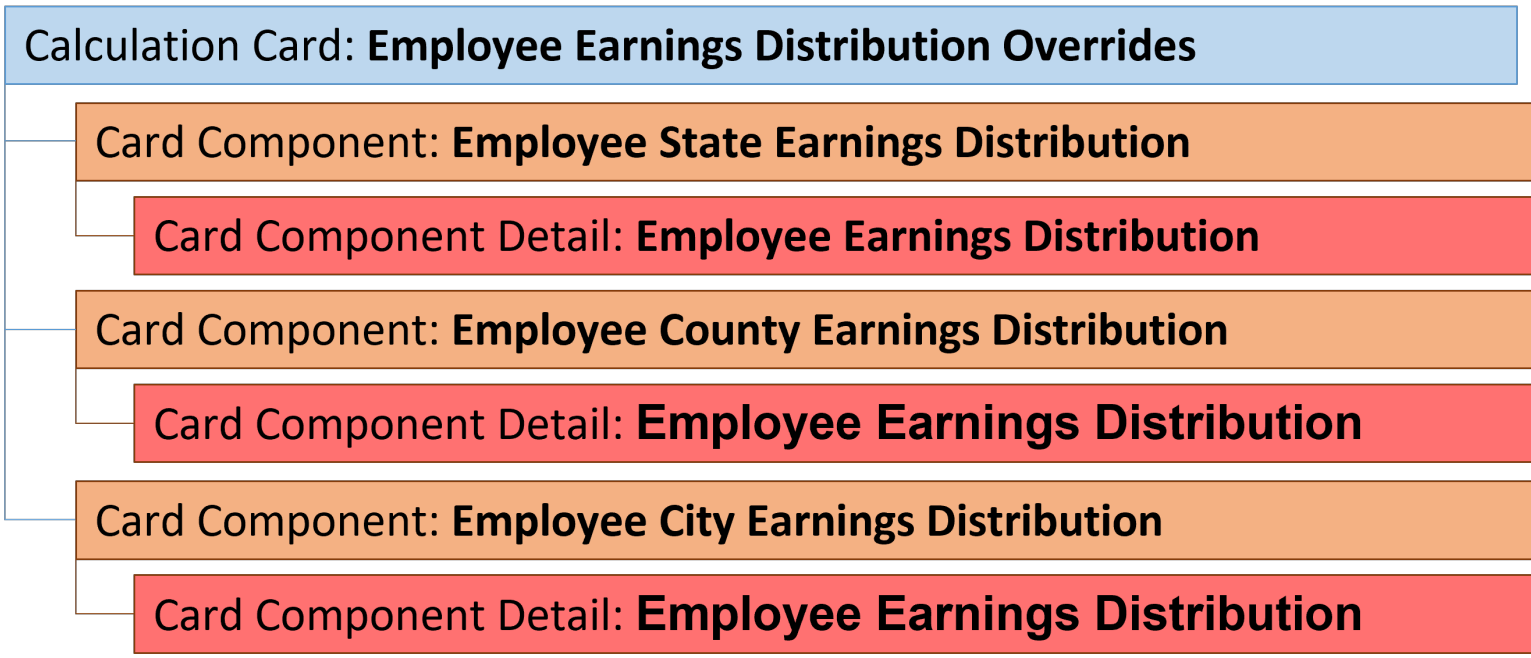
The Employee Earnings Distribution Overrides calculation card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements. The Employee Earnings Distribution Overrides card utilizes the following record types:

Record Types for Employee Earnings Distribution Overrides Card

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Supply a component detail record for each flexfield context required by each card component.	ComponentDetail

Employee Earnings Distribution Overrides Card Hierarchy

The hierarchy of Calculation Card components applicable **to Employee Earnings Distribution Overrides** is described in this diagram, which includes the state, county and city component groups:



Guidelines for Loading Employee Earnings Distribution Overrides Card

Create an Employee Earnings Distribution Overrides card for any overriding jurisdictions, for each payroll relationship the employee is in.

Even if you are updating an existing calculation card and the calculation card itself isn't being updated, still include the calculation card record to group other related data supplied in the file.

Employee Earnings Distribution Overrides Calculation Card Attributes

The Employee Earnings Distribution Overrides calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Employee Earnings Distribution Overrides calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner		The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName		The name of the legislative data group for the calculation card.
DirCardDefinitionName		The name of the card definition. Specify 'Employee Earnings Distribution Overrides'.
CardSeqeunce		Specify '1'. Not required when source keys are used.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate		The start date of the calculation card.

These attributes are supplied against the CalculationCard file discriminator.

Guidelines for Loading Employee State Earnings Distribution Card Components

The Employee State Earnings Distribution card component is used to capture the percentage of earnings distribution that applies to the state.

Card Component Hierarchy

The Employee State Earnings Distribution card component utilizes the Employee Earnings Distribution flexfield context and data for this is loaded using the Component Detail record type.

Card Component Attributes

The Employee State Earnings Distribution card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee State Earnings Distribution card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent **Employee Earnings Distribution Overrides** calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the Employee State Earnings Distribution card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Earnings Distribution Overrides calculation card. If updating an existing Employee State Earnings Distribution card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Specify 'Employee State Earnings Distribution'.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.
Context1	N/A	The geocode number that identifies the state.

Component Detail Attributes

The component detail record type allows you to upload data into flexfield segments.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Employee Earnings Distribution (ORA_HRX_US_EARN_DISTRIBUTION)

You can find the flexfield segment attribute name for this flexfield context using the **View Business Objects** task.

Core attributes for Component Details:

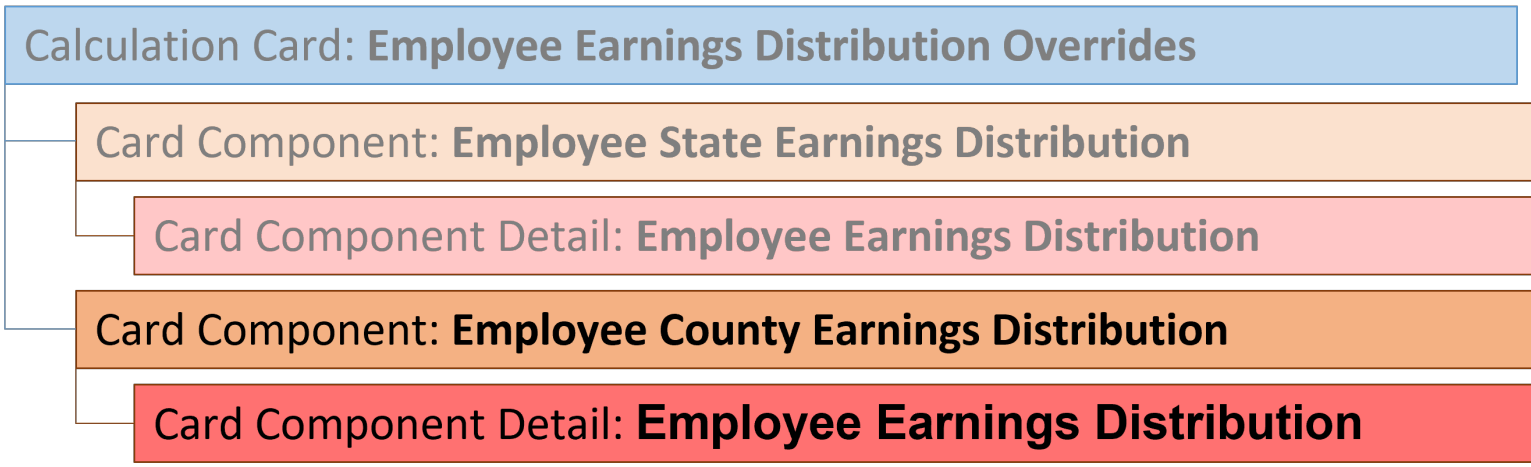
HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the Employee Earnings Distribution component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **Employee State Earnings Distribution** card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee State Earnings Distribution card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component. Supply 'Employee State Earnings Distribution'.
DirInformationCategory	N/A	The code for the flexfield context. Supply 'ORA_HRX_US_EARN_DISTRIBUTION'.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Note: When supplying values to lookup validated flexfield segments, you can provide the lookup meaning using the '_Display' suffixed attribute name, but it's recommended that you supply the lookup code to remove potential translation issues.

Guidelines for Loading Employee County Earnings Distribution Card Components

The Employee County Earnings Distribution card component is used to capture the percentage of earnings distribution that applies to the county.

Card Component Hierarchy



The **Employee County Earnings Distribution** card component utilizes the Employee Earnings Distribution flexfield context and data for this is loaded using the Component Detail record type.

It references the **Employee State Earnings Distribution** card component as its parent card component.

Card Component Attributes

The Employee County Earnings Distribution card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee County Earnings Distribution card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent **Employee Earnings Distribution Overrides** calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
ParentDirCardCompId(Source SystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	The parent Employee State Earnings Distribution card component. When using source keys, supply this attribute with the value supplied to the SourceSystemId on the parent Employee State Earnings Distribution card component.
EffectiveStartDate	N/A	The start date of the Employee County Earnings Distribution card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Earnings Distribution Overrides calculation card. If updating an existing Employee County Earnings Distribution card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Specify 'Employee County Earnings Distribution'.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.
Context1	N/A	The geocode number that identifies the state.
Context2	N/A	The geocode number that identifies the county.

Component Detail Attributes

The component detail record type allows you to upload data into flexfield segments.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Employee Earnings Distribution (ORA_HRX_US_EARN_DISTRIBUTION)

You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Core attributes for Component Details:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory,	A unique identifier for the Employee Earnings Distribution component detail. For new

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **Employee County Earnings Distribution** card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee County Earnings Distribution card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component. Supply 'Employee County Earnings Distribution'.
DirInformationCategory	N/A	The code for the flexfield context. Supply 'ORA_HRX_US_EARN_DISTRIBUTION'.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Note: When supplying values to lookup validated flexfield segments, you can provide the lookup meaning using the '_Display' suffixed attribute name, but it's recommended that you supply the lookup code to remove potential translation issues.

Guidelines for Loading Employee City Earnings Distribution Card Components

The Employee City Earnings Distribution card component is used to capture the percentage of earnings distribution that applies to the city.

Card Component Hierarchy

The **Employee City Earnings Distribution** card component utilizes the Employee Earnings Distribution flexfield context and data for this is loaded using the Component Detail record type.

It references the **Employee County Earnings Distribution** card component as its parent card component.

Card Component Attributes

The Employee City Earnings Distribution card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee City Earnings Distribution card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent **Employee Earnings Distribution Overrides** calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
ParentDirCardCompId(Source SystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	The parent Employee County Earnings Distribution card component. When using source keys, supply this attribute with the value supplied to the SourceSystemId on the parent Employee County Earnings Distribution card component.
EffectiveStartDate	N/A	The start date of the Employee City Earnings Distribution card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Earnings Distribution Overrides calculation card. If updating an existing Employee City Earnings Distribution card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Employee City Earnings Distribution'.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.
Context1	N/A	The geocode number that identifies the state.
Context2	N/A	The geocode number that identifies the county.
Context3	N/A	The geocode number that identifies the city.

Component Detail Attributes

The component detail record type enables you to upload data into flexfield segments.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Employee Earnings Distribution (ORA_HRX_US_EARN_DISTRIBUTION)

You can find the flexfield segment attribute name for this flexfield context using the **View Business Objects** task.

Core attributes for Component Details:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName,	A unique identifier for the Employee Earnings Distribution component detail. For new component detail records supply the source

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **Employee City Earnings Distribution** card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee City Earnings Distribution card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component. Supply 'Employee City Earnings Distribution'.
DirInformationCategory	N/A	The code for the flexfield context. Supply 'ORA_HRX_US_EARN_DISTRIBUTION'.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Note: When supplying values to lookup validated flexfield segments, you can provide the lookup meaning using the '_Display' suffixed attribute name, but it's recommended that you supply the lookup code to remove potential translation issues.

Example of Loading Employee Earnings Distribution Overrides Card

Here's an example to load the employee earnings distribution overrides card.

This example creates an Employee Earnings Distribution Overrides card for assignment E91817, staring on 1-Jan-2022.

A value of 50% is supplied for each of the State, County and City component cards.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
CardSequence|EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|E91817_EEDO|US LDG|Employee Earnings Distribution Overrides|1|2022/01/01|E91817

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardCompDefName|
ComponentSequence|EffectiveStartDate|DirCardId(SourceSystemId)|ParentDirCardCompId(SourceSystemId)|Context1|
Context2|Context3
MERGE|CardComponent|VISION|E91817_ST_DO|US LDG|Employee State Earnings Distribution|1|2022/01/01|
E91817_EEDO||1||
MERGE|CardComponent|VISION|E91817_CO_DO|US LDG|Employee County Earnings Distribution|2|2022/01/01|
E91817_EEDO|E91817_ST_DO|1|3|
MERGE|CardComponent|VISION|E91817_CI_DO|US LDG|Employee City Earnings Distribution|3|2022/01/01|E91817_EEDO|
E91817_CO_DO|1|3|3826

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardCompDefName|
ComponentSequence|EffectiveStartDate|DirCardCompId(SourceSystemId)|DirInformationCategory|FLEX:Deduction
 Developer DF|_ORA_PERCENTAGE(Deduction Developer DF=ORA_HRX_US_EARN_DISTRIBUTION)
MERGE|ComponentDetail|VISION|E91817_ST_DO|US LDG|Employee State Earnings Distribution|1|2022/01/01|
E91817_ST_DO|ORA_HRX_US_EARN_DISTRIBUTION|ORA_HRX_US_EARN_DISTRIBUTION|50
MERGE|ComponentDetail|VISION|E91817_CO_DO|US LDG|Employee County Earnings Distribution|2|2022/01/01|
E91817_CO_DO|ORA_HRX_US_EARN_DISTRIBUTION|ORA_HRX_US_EARN_DISTRIBUTION|50
MERGE|ComponentDetail|VISION|E91817_CI_DO|US LDG|Employee City Earnings Distribution|3|2022/01/01|
E91817_CI_DO|ORA_HRX_US_EARN_DISTRIBUTION|ORA_HRX_US_EARN_DISTRIBUTION|50
```

Related Topics

- Overview of Employee Earnings Distribution Overrides Card
- Guidelines for Loading Employee Earnings Distribution Overrides Card
- Guidelines for Loading Employee State Earnings Distribution Card Components
- Guidelines for Loading Employee County Earnings Distribution Card Components
- Guidelines for Loading Employee City Earnings Distribution Card Components

Employee Reporting Information

Overview of US Reporting Information Cards

The Reporting Information card stores information that's required for the Quarterly and Year-end reporting. The card allows you to update information on an employee's or retiree's record.

Reporting Information Card Component	Component Fields
Reporting Information (Federal)	Legal Representative Corporate Officer Eligible for Retirement Plan Reporting Location for Work-at-Home Employees Probationary Code

Reporting Information Card Component	Component Fields
	Family member with majority interest Third-Party Interfaces • ADP Special Processing • ADP Special Sort Code
State Reporting Information	Alaska: • Geographic Code • Occupational Code California: • Wage Plan Code Colorado: • Seasonal Indicator Indiana: • Seasonal Worker • Taxation Location Maine: • Seasonal Worker • Wage Plan Code Missouri: • Seasonal Worker New York: • Part Year Resident North Carolina: • Seasonal Worker Pennsylvania: • Wage Plan Code Puerto Rico: • Domestic Services • Others • Services rendered by a qualified physician under Act 14-2017 Vermont: • Health care status

Reporting Information Card Component	Component Fields
Regional Tax Information	Puerto Rico: • W2 control number • W2C control number
Retiree Reporting Information	Account Number Annuity Percentage FATCA Filing Requirement First Year of Designated Roth Contribution Taxable Amount Not Determined Total Distribution Total Distribution Percentage Traditional, Simplified Employee Pension (SEP), or Savings Incentive Match for Employees (SIMPLE) Individual Retirement Account (IRA)

Considerations and Prerequisites

For employee Reporting Information cards

The values entered on the Reporting Information Calculation Card are passed on the quarterly tax file and update the employee for year-end reporting. You need to create a Federal and Regional calculation component that includes the proper regional association for the state or territory that the person is employed and must set the appropriate fields for each.

For Retiree Reporting Information cards

A retiree's tax card and TRU association has a one-to-one relationship with the Reporting Information card. For every tax card TRU association, there must be a corresponding Retiree Reporting Information component. When you create a TRU association on the tax card, the association process automatically creates a Reporting Information card with a Retiree Reporting Information component associated to the same TRU.

Reporting Information Card Record Types

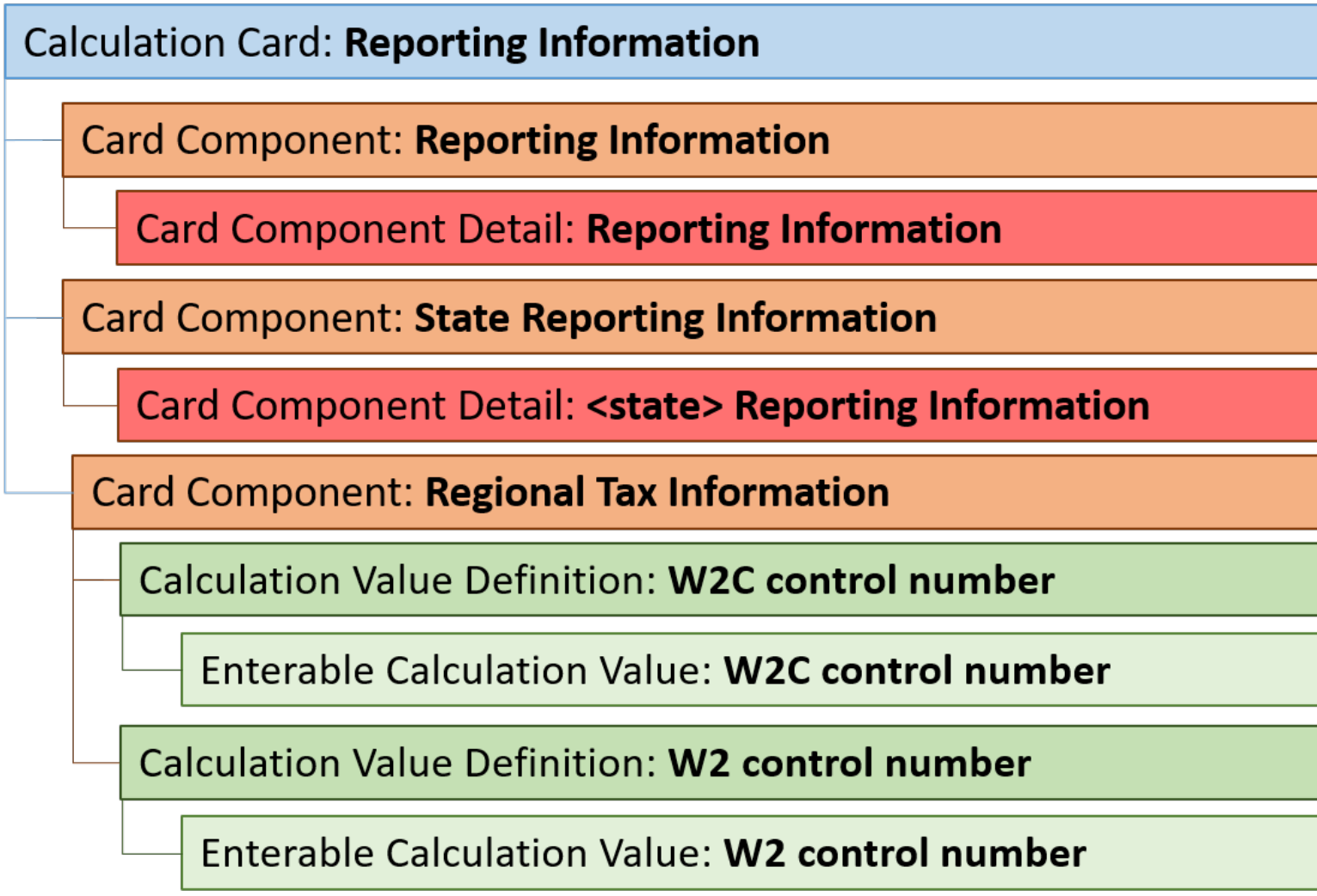
The Reporting Information calculation card utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections	CardComponent

Component	Functional Description	File Discriminator
	describe the card components applicable to this calculation card and the child records that are required for each card component.	
Component Detail	Supply a component detail record for each flexfield context required by each card component.	ComponentDetail
Calculation Value Definition	Allows the creation of value definitions so that overriding values can be specified on the card component. Details of the specific value definitions are provided in the following sections.	CalculationValueDefinition
Enterable Calculation Value	Used to specify an overriding value for each calculation value definition.	EnterableCalculationValue
Card Association	Associates the calculation card with the Tax Reporting Unit the employee or retiree reports to.	CardAssociation
Card Association Detail		CardAssociationDetails

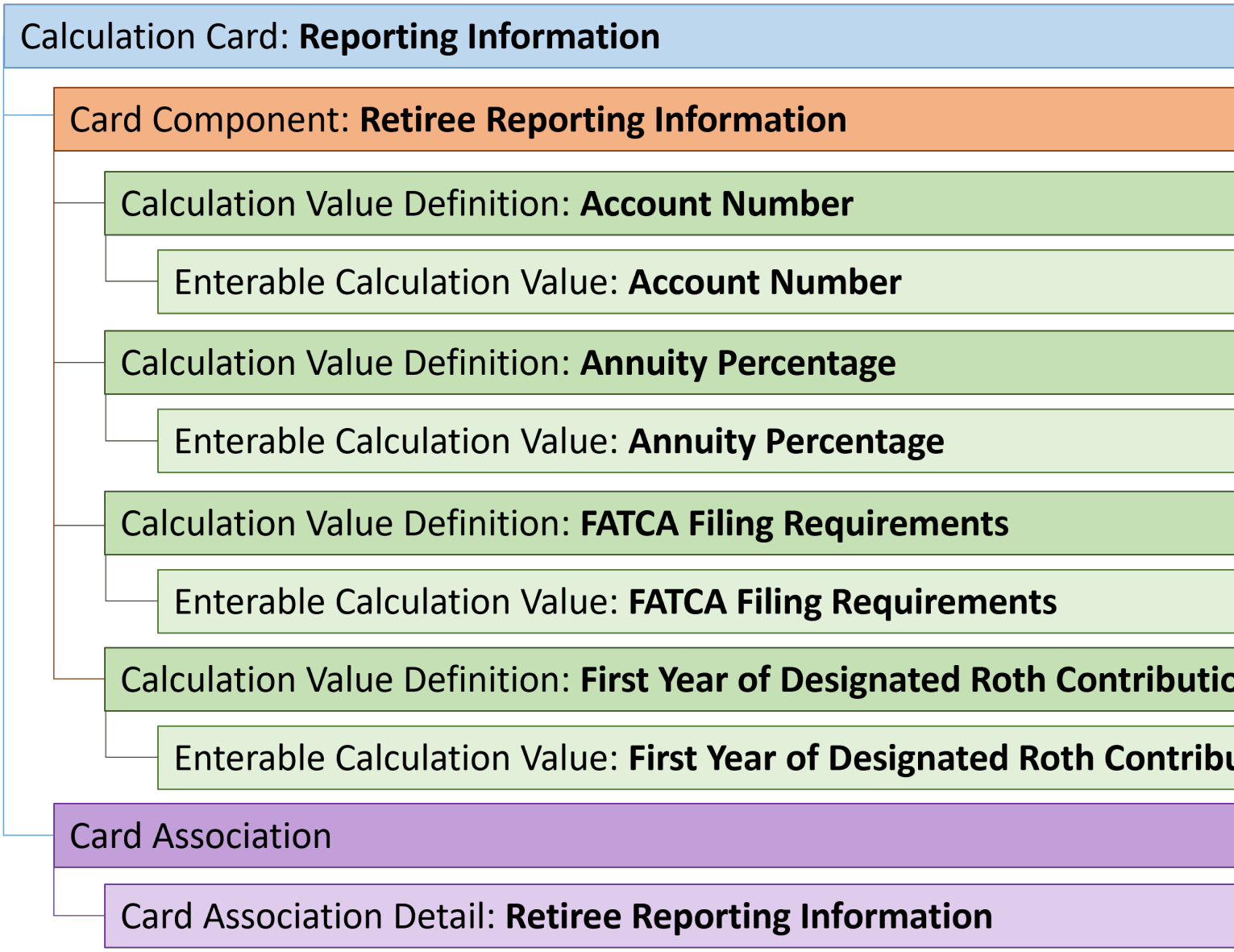
Reporting Information for Employees Card Hierarchy

The hierarchy of Calculation Card components applicable to employee Reporting Cards are described in this diagram:



Retiree Reporting Information Card Hierarchy

The hierarchy of Calculation Card components applicable to Retiree Reporting Cards are described in this diagram:



Refer to the help topics in this section for guidelines and examples of creating Reporting Information cards.

Guidelines for Loading Employee Reporting Information Cards for the US

Create Reporting Information Calculation Cards for every employee you're maintaining Quarterly and Year-end tax data for.

Even if you're updating an existing calculation card and the calculation card itself isn't being updated, still include the calculation card record to group other related data supplied in the file.

Reporting Information Calculation Card Attributes

The Reporting Information calculation card for employees uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Reporting Information calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Reporting Information'.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card.

Guidelines for Loading Reporting Information Card Components for US Employees

The employee federal reporting data is captured in the Reporting Information card component, within the Reporting Information calculation card.

Reporting Information Card Component Hierarchy

Card Component Attributes for Reporting Information

The Reporting Information Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Reporting Information card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent **Reporting Information** calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the Reporting Information card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Reporting Information calculation card. If updating an existing Reporting Information card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Specify 'Reporting Information'.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.

These attributes are supplied against the CardComponent file discriminator and must be supplied along with a CalculationCard record for the parent **Reporting Information** card.

Component Detail Attributes for Reporting Information

The component detail record type allows you to upload data into flexfield segments.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

– Reporting Information (HRX_US_REP_REL)

You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

Core attributes for Component Details:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the Reporting Information component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **Reporting Information** card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Reporting Information card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context. Supply 'HRX_US_REP_REL'.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Note: When supplying values to lookup validated flexfield segments, you can provide the lookup meaning using the '_Display' suffixed attribute name, but it is recommended that you supply the lookup code to remove potential translation issues.

Example of Loading Reporting Information Cards for US Employees

This example creates the employee federal data to the Reporting Information calculation card, for employee E1762782 on 01-January-2022.

Source keys are used to identify all record types.

The CalculationCard.dat file is used to bulk-load Reporting Information cards with HCM Data Loader.

Descriptive Flexfield Code: Deduction Developer DF

Segment Code: US_ELIGIBLE_FOR_PRETAX_CATCHUP

Prompt: Eligible for pre-tax age catch-up

Value Set: HRC_YES_NO

Default Value = Y if left blank

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardDefinitionName|AssignmentNumber|CardSequence
MERGE|CalculationCard|VISION|RI_1762782E|USA LDG|2022/01/01|Reporting Information|E1762782|1

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardId(SourceSystemId)|DirCardCompDefName
MERGE|CardComponent|VISION|RI_1762782E_RI|USA LDG|2022/01/01|RI_1762782E|Reporting Information

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardCompId(SourceSystemId)|DirCardCompDefName|DirInformationCategory|FLEX:Deduction Developer
 DF|_LEGAL_REPRESENTATIVE(Deduction Developer DF=HRX_US_REP_REL)|_CORPORATE_OFFICER(Deduction
 Developer DF=HRX_US_REP_REL)|_ELIGIBLE_FOR_RETIREMENT_PLAN(Deduction Developer DF=HRX_US_REP_REL)|
_PROBATIONARY_CODE(Deduction Developer DF=HRX_US_REP_REL)|_REP_LOC_FOR_WORK_AT_HOME_Display(Deduction
 Developer DF=HRX_US_REP_REL)|_FAMILY_MAJORITY_INTEREST(Deduction Developer DF=HRX_US_REP_REL)|
_US_ELIGIBLE_FOR_PRETAX_CATCHUP(Deduction Developer DF=HRX_US_REP_REL)
MERGE|ComponentDetail|VISION|RI_1762782E_USREPREL|USA LDG|2022/01/01|RI_1762782E_RI|Reporting Information|
HRX_US_REP_REL|HRX_US_REP_REL|Y|Y|Y|1|Redwood Shores San Mateo US|Y|Y
```

The card component record identifies the parent calculation card using the DirCardId(SourceSystemId) attribute which takes the same value as the SourceSystemId attribute on the parent CalculationCard record.

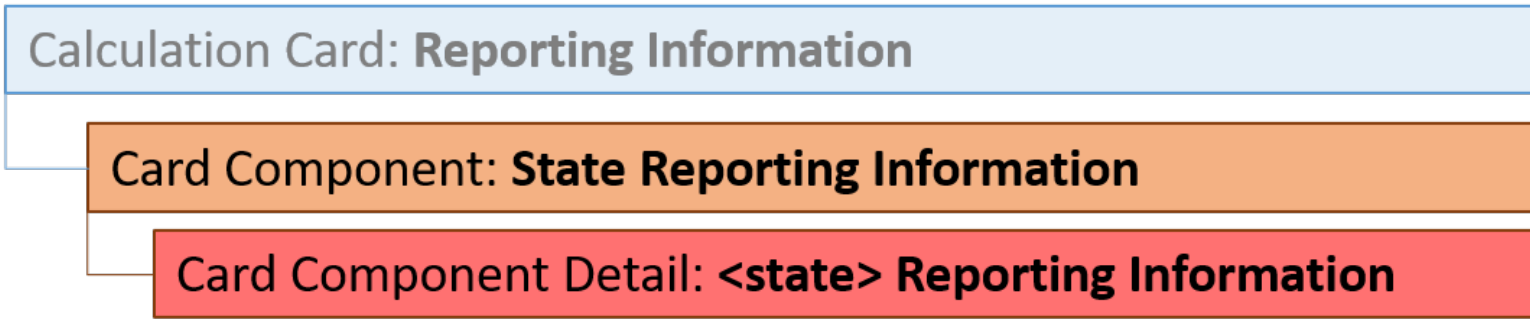
The component detail record identifies the parent card component using the DirCardCompID(SourceSystemId) attribute.

Refer to the Template: BI Publisher Report - United States: Reporting Location for Work-at-Home Employees topic in Cloud Customer Connect for a report that extracts the reporting location and Fusion surrogate ID for work-at-home employees. This is needed when supplying the 'Reporting Location for Work-at-Home Employees' flexfield segment on the Reporting Information card component.

Guidelines for Loading State Reporting Information Card Components for US Employees

Some US states require additional reporting information. This is captured in the State Reporting Information card component, within the Reporting Information calculation card.

State Reporting Information Card Component Hierarchy



Card Component Attributes for State Reporting Information

The State Reporting Information Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the StateReporting Information card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent **Reporting Information** calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the State Reporting Information card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Reporting Information calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		If updating an existing State Reporting Information card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Specify 'Reporting Information'.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.
Context1	N/A	The geocode that identifies the US state. Refer to the Cloud Customer Connect topic *Template: BI Publisher Report: United States - State, County and City Geography Codes*.

These attributes are supplied against the CardComponent file discriminator and must be supplied along with a CalculationCard record for the parent **Reporting Information** card.

Component Detail Attributes for State Reporting Information

The component detail record type allows you to upload data into flexfield segments.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context that pertains the state:

- Alaska Reporting Information (HRX_US_REP_REL_STATE_AK)
- California Reporting Information (HRX_US_REP_REL_STATE_CA)
- Colorado Reporting Information (HRX_US_REP_REL_STATE_CO)
- Indiana Reporting Information (HRX_US_REP_REL_STATE_IN)
- Maine Reporting Information (HRX_US_REP_REL_STATE_ME)
- Missouri Reporting Information (HRX_US_REP_REL_STATE_MO)
- New Jersey Reporting Information (HRX_US_REP_REL_STATE_NJ)
- North Carolina Reporting Information (HRX_US_REP_REL_STATE_NC)
- Pennsylvania Reporting Information (HRX_US_REP_REL_STATE_PA)
- Vermont Reporting Information (HRX_US_REP_REL_STATE_VT)

A separate Component Detail file line should be included for each flexfield context you are supplying data for.

> **Note:** You can find the flexfield segment attribute name for this flexfield context using the View Business Objects task.

Core attributes for Component Details:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the StateReporting Information component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **State Reporting Information** card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the StateReporting Information card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_US_REP_REL_STATE_AK'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Note: When supplying values to lookup validated flexfield segments, you can provide the lookup meaning using the '_Display' suffixed attribute name, but it is recommended that you supply the lookup code to remove potential translation issues.

Example of Loading State Reporting Information Card Component for US Employees

This example uploads the employee state data for California into the Reporting Information calculation card for employee assignment E8177672 on the 01-January-2018.

Source keys are used to identify all record types.

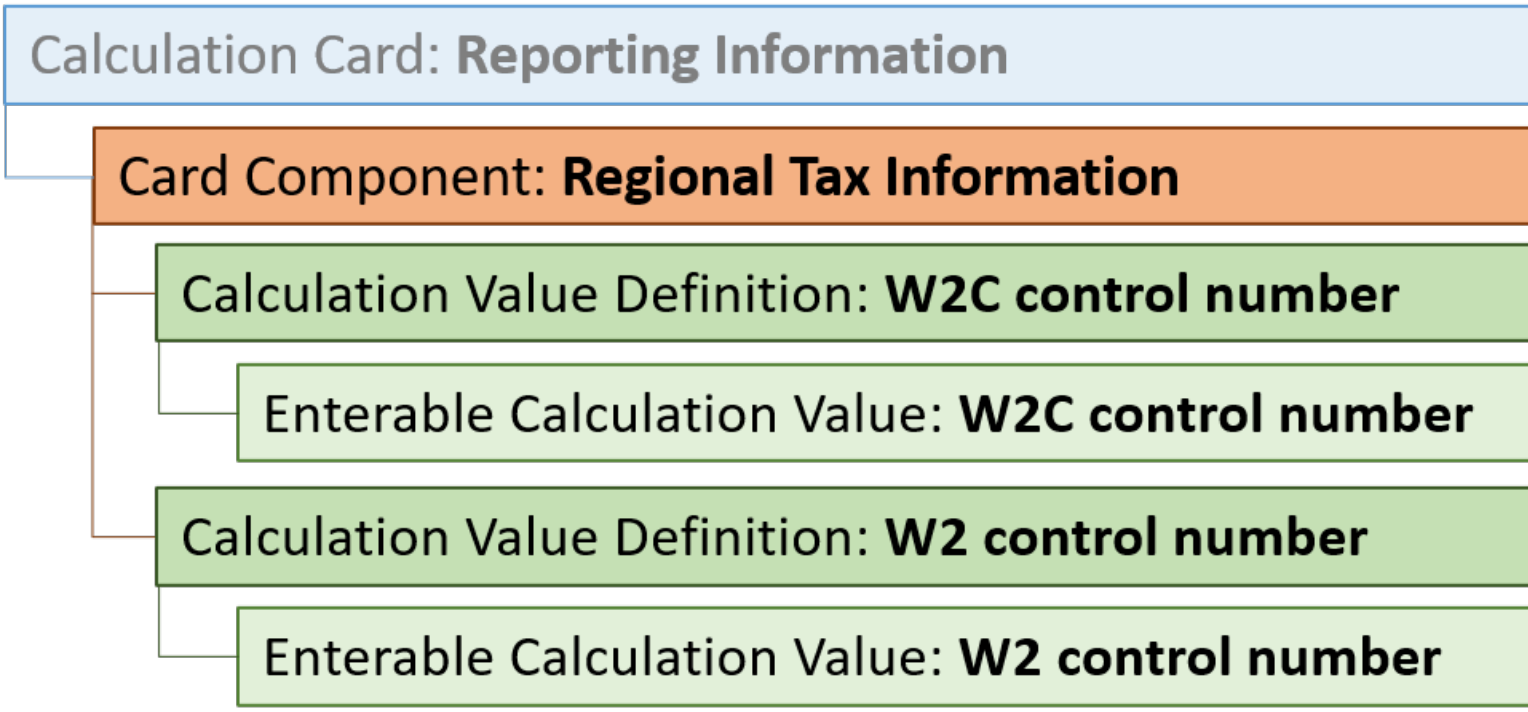
The CalculationCard.dat file is used to bulk-load Reporting Information cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardDefinitionName|AssignmentNumber|CardSequence
MERGE|CalculationCard|VISION|RI_8177672E|US LDG|2018/01/01|Reporting Information|E8177672|1

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardId(SourceSystemId)|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|RI_SRI_8177672E|US LDG|2018/01/01|RI_8177672E|State Reporting Information|5

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardCompId(SourceSystemId)|DirCardCompDefName|DirInformationCategory|FLEX:Deduction Developer DF|
_WAGE_PLAN_CODE(Deduction Developer DF=HRX_US_REP_REL_STATE_CA)
MERGE|ComponentDetail|VISION|RI_8177672E_STATE_CA|US LDG|2018/01/01|RI_SRI_8177672E|State Reporting
 Information|HRX_US_REP_REL_STATE_CA|HRX_US_REP_REL_STATE_CA|S
```

Guidelines for Loading Regional Tax Information Card Components for US Employees

The Puerto Rico control numbers are recorded in the Regional Tax Information card component, within the Reporting Information card.

Regional Tax Information Card Component Hierarchy



The Regional Tax Information card component has two value definitions to record the Puerto Rico control numbers. The following sections describe how to supply valid file lines for these record types.

Card Component Attributes for Regional Tax Information

The Regional Tax Information card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Regional Tax Information card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Reporting Information calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Regional Tax Information card component. This must be after or same

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		as the EffectiveStartDate on the Reporting Information calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component The component definition name. Provide the name of the component as generated by the element template.
DirCardCompDefName	N/A	The component definition name. Provide the name of the component as generated by the element template
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.
Context1	N/A	The geocode that identifies the Puerto Rico state, specify '72'.
Context2	N/A	The surrogate ID of the tax reporting unit. Refer to the Cloud Customer Connect topic *Template: BI Publisher Report: Tax Reporting Units by Legislation*.

These attributes are supplied against the CardComponent file discriminator and must be supplied along with a CalculationCard record for the parent **Reporting Information** card.

Puerto Rico Control Number Value Definitions

The Regional Tax Information card component uses these value definitions to supply override values for Puerto Rico Control Numbers:

- W2C control number
- W2 control number

Calculation Value Definition Attributes for Regional Tax Information

The Calculation Value Definition record type specifies the name of the value definition you are supplying an override value for.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Regional Tax Information card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent Regional Tax Information card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent Regional Tax Information card component.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.
Context1	N/A	The geocode that identifies the Puerto Rico state, specify '72'.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent **Regional Tax Information** card component and a CalculationCard record for the owning **Reporting Information** card.

Enterable Calculation Value Attributes for Regional Tax Information

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName,	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes.
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Example of Loading Puerto Rico Control Numbers

This example loads the Puerto Rico control numbers into the Reporting Information calculation card for employee assignment E1404348 on 01-January-2018.

The CalculationCard.dat file is used to bulk-load Reporting Information cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardDefinitionName|AssignmentNumber
MERGE|CalculationCard|VISION|RI_1404348|US LDG|2018/01/01|Reporting Information|E1404348

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
DirCardCompDefName|DirCardId(SourceSystemId)|Context1|Context2
MERGE|CardComponent|VISION|RTI_1404348|US LDG|2018/01/01|Regional Tax Information|RI_1404348|72|
300100095766624

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
EffectiveStartDate|DirCardCompDefName|SourceId(SourceSystemId)|ValueDefinitionName|Context1
MERGE|CalculationValueDefinition|VISION|RTI_1404348_WC2|US LDG|2018/01/01|Regional Tax Information|
RTI_1404348|W2C control number|72
MERGE|CalculationValueDefinition|VISION|RTI_1404348_WC|US LDG|2018/01/01|Regional Tax Information|
RTI_1404348|W2 control number|72

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
EffectiveStartDate|ValueDefnId(SourceSystemId)|Value1
MERGE|EnterableCalculationValue|VISION|RTI_1404348_WC2|US LDG|2018/01/01|RTI_1404348_WC2|12354689
MERGE|EnterableCalculationValue|VISION|RTI_1404348_WC|US LDG|2018/01/01|RTI_1404348_WC|123456789
```

Guidelines for Loading Retiree Reporting Information Cards for the US

The Retiree Reporting Information component on a retiree's Reporting Information card provides optional information for your Forms 1099-R.

Create a Calculation Card record for every Retiree you are maintaining data for Form 1099-R. Even if you are updating an existing calculation card and the calculation card itself is not being updated, still include the calculation card record to group other related data supplied in the file.

Reporting Information Calculation Card Attributes

The Reporting Information calculation card for retirees uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Reporting Information calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Reporting Information'
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card.

These attributes are supplied against the CalculationCard file discriminator.

Card Component Attributes for Reporting Information

The Retiree Reporting Information Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName,	A unique identifier for the RetireeReporting Information card component. For new card

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, DirCardCompDefName	components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent **Reporting Information** calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Retiree Reporting Information card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Reporting Information calculation card. If updating an existing Retiree Reporting Information card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Specify 'Retiree Reporting Information'.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.
Context1	N/A	The surrogate ID of the tax reporting unit for the retiree. Refer to the Cloud Customer Connect topic Template: BI Publisher Report: Tax Reporting Units by Legislation for a report that extracts the tax reporting units and their surrogate ID values for the specified legislation.

These attributes are supplied against the CardComponent file discriminator and must be supplied along with a CalculationCard record for the parent Reporting Information card.

Value Definitions for Retiree Reporting Information

This table lists the value definitions applicable to the Retiree Reporting Information card component.

Value Definition Name	Functional Description
Account Number	The payers account number for the payee. Enter a number that can be used as a unique identifier on the original payee 1099-R.
Annuity Percentage	The percentage will be entered if an annuity contract is part of a multiple recipient lump-sum distribution. Specify the value as a number, such as 10 for 10%.
FATCA Filing Requirements	The FATCA Filing requirement is for when banks issue 1099-INT forms. Specify Y or N.
First Year of Designated Roth Contribution	The first year a designated Roth contribution was made. Specify the four-digit year, such as 2018.
Taxable Amount not Determined	The payer was unable to determine the taxable amount of the employee's distribution. Specify Y or N.
Total Distribution	The distribution is for a total distribution. Specify Y or N.
Total Distribution Percentage	The percentage of the total distribution if split between more than one person. Specify the value as a number, such as 10 for 10%.
Traditional, SEP or SIMPLE IRA	The employee is taking regular payments from a traditional, Sep, or SIMPLE IRA. Specify Y or N.

Calculation Value Definition Attributes

The Calculation Value Definition record type specifies the name of the value definition you are supplying an override value for. The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Retiree Reporting Information card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent Retiree Reporting Information card component or the date the calculation value definition starts, if later.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
DirCardCompDefName	N/A	The definition name of the parent Retiree Reporting Information card component.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent Regional Tax Information card component and a CalculationCard record for the owning Reporting Information card.

Enterable Calculation Value Attributes for Regional Tax Information

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	AssignmentNumber	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes.
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

Reporting Information Card Association Attributes

The card association record should be supplied along with the Calculation Card record for the Reporting Information card the association is for.

The Card Association uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Reporting Information card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent **Reporting Information** calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Reporting Information calculation card.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Tax Withholding calculation card.

These attributes are supplied against the CardAssociation discriminator.

Card Association Details

The card association details record associates the Retiree Reporting Information card component with the retiree's assignments.

If the retiree has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Identify the Retiree Reporting Information card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

These attributes are supplied against the CardAssociationDetail discriminator.

Related Topics

- Guidelines for Loading Calculation Cards

Example of Loading Retiree Reporting Information Cards for the US

This example create the Retiree Reporting Information calculation card for retiree assignment N1254332 on the 01-January-2020.

The CalculationCard.dat file is used to bulk-load Reporting Information cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|RI_1254332|US LDG|Reporting Information|2020/01/01|N1254332

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardCompDefName|
EffectiveStartDate|DirCardId(SourceSystemId)|Context1
```

```
MERGE|CardComponent|VISION|RI_1254332_RII|US LDG|Retiree Reporting Information|2020/01/01|RI_1254332|
300100193871203

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|RI_1254332_RII_VD1|US LDG|RI_1254332_RII|2020/01/01|Retiree
 Reporting Information|Account Number
MERGE|CalculationValueDefinition|VISION|RI_1254332_RII_VD2|US LDG|RI_1254332_RII|2020/01/01|Retiree
 Reporting Information|Annuity Percentage
MERGE|CalculationValueDefinition|VISION|RI_1254332_RII_VD3|US LDG|RI_1254332_RII|2020/01/01|Retiree
 Reporting Information|FATCA Filing Requirements
MERGE|CalculationValueDefinition|VISION|RI_1254332_RII_VD4|US LDG|RI_1254332_RII|2020/01/01|Retiree
 Reporting Information|First Year of Designated Roth Contribution
MERGE|CalculationValueDefinition|VISION|RI_1254332_RII_VD5|US LDG|RI_1254332_RII|2020/01/01|Retiree
 Reporting Information|Taxable Amount not Determined
MERGE|CalculationValueDefinition|VISION|RI_1254332_RII_VD6|US LDG|RI_1254332_RII|2020/01/01|Retiree
 Reporting Information|Total Distribution
MERGE|CalculationValueDefinition|VISION|RI_1254332_RII_VD7|US LDG|RI_1254332_RII|2020/01/01|Retiree
 Reporting Information|Total Distribution Percentage
MERGE|CalculationValueDefinition|VISION|RI_1254332_RII_VD8|US LDG|RI_1254332_RII|2020/01/01|Retiree
 Reporting Information|Traditional, SEP or SIMPLE IRA

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|VISION|RI_1254332_RII_VD1|US LDG|RI_1254332_RII_VD1|2020/01/01|Account
 Number|125364789
MERGE|EnterableCalculationValue|VISION|RI_1254332_RII_VD2|US LDG|RI_1254332_RII_VD2|2020/01/01|Annuity
 Percentage|10
MERGE|EnterableCalculationValue|VISION|RI_1254332_RII_VD3|US LDG|RI_1254332_RII_VD3|2020/01/01|FATCA Filing
 Requirements|N
MERGE|EnterableCalculationValue|VISION|RI_1254332_RII_VD4|US LDG|RI_1254332_RII_VD4|2020/01/01|First Year of
 Designated Roth Contribution|2015
MERGE|EnterableCalculationValue|VISION|RI_1254332_RII_VD5|US LDG|RI_1254332_RII_VD5|2020/01/01|Taxable
 Amount not Determined|N
MERGE|EnterableCalculationValue|VISION|RI_1254332_RII_VD6|US LDG|RI_1254332_RII_VD6|2020/01/01|Total
 Distribution|N
MERGE|EnterableCalculationValue|VISION|RI_1254332_RII_VD7|US LDG|RI_1254332_RII_VD7|2020/01/01|Total
 Distribution Percentage|10
MERGE|EnterableCalculationValue|VISION|RI_1254332_RII_VD8|US LDG|RI_1254332_RII_VD8|2020/01/01|Traditional,
 SEP or SIMPLE IRA|N

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|RI_1254332_TRU|US LDG|RI_1254332|2020/01/01|US TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirRepCardId(SourceSystemId)|EffectiveStartDate|DirCardCompId(SourceSystemId)|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|RI_1254332_TRU_RII|US LDG|RI_1254332_TRU|2020/01/01|RI_1254332_RII|
N1254332
```

Initialize and Adjust Balances

Overview of Balance Initialization for the US

The US Legislative Balances are balances that Oracle Payroll Cloud uses to perform accurate tax reporting and payroll tax calculations, such as for W-2 reporting and State Quarterly Wage Listings.

Whether they're required or not, that's determined at the individual employee level. For example, if an employee doesn't participate in the company 401 (k) plan, the Deferred Compensation 401k balance isn't needed for that employee.

For a list of the balances to load, supported dimensions and the process overview, refer to the document below in the US Information Center.

US Information Center

https://support.oracle.com/rs?type=doc&id=2063588.2

US – Payroll tab > Product Documentation > Technical Briefs > Balance Initialization

US Legislative or User-defined Context Columns

There are six legislative or user-defined contexts, which must be populated if the balance dimension expects a value for these contexts. These contextual values are used because columns are not available to populate legislative or user-defined data. Instead, they require entry in the user-defined context columns as outlined below. For each context, there is a context name and value attribute. For example:

- ContextOneName
- ContextOneValue

Populate the context name attributes with the name of the context usage for that legislative or user-defined context, and the context value attributes with the actual value of that context.

For example, in case the Child Support involuntary deduction balance for the Relationship Tax Unit, Third Party Payee and Reference Code Year to Date dimension requires an additional context, in order to define the child support reference code, you populate these values:

Attribute	Value
ContextOneName	Reference Code
ContextOneValue	123456

If the context you're populating is a lookup, populate the context fields in the following manner:

- Context Name: Populate with the lookup type meaning
- Context Value: Populate with the lookup code (not the lookup type meaning)

Let's take another example. If you need to populate the Context One Name and Context One Value fields for the Relationship Tax Unit, Area1, Tax Unit2, Area5,6 Year to Date dimension for PA Act 32 balances, you would populate these values:

Attribute	Value
ContextOneName	Area5
ContextOneValue	700102
ContextTwoName	Area6
ContextTwoValue	220401

Attribute	Value
ContextThreeName	Tax Unit 2
ContextThreeValue	22

These legislative contexts are required only if the employee's work or residence is in a jurisdiction subject to Pennsylvania Act 32.

The meanings of the additional PA Act 32 contexts are:

- Area5 = Resident Political Subdivision (PSD) Code
- Area6 = Work PSD Code
- Tax Unit 2 = Tax Collection District Code

Note: If you're initializing involuntary deduction balances, you must create the Involuntary Deductions card before processing the balance initialization.

Area attributes for Initializing US Balances

Along with the core attributes, details of which are available in the View Business Objects task, there are area attributes that may be relevant for your US balances:

HCM Data Loader Attribute	Functional Description
AreaOne	State information for the balance value. Supply the geocode for the state. For more information, refer to *Template: BI Publisher Report: United States - State, County and City Geography Codes*
AreaTwo	County information for the balance value. Supply the geocode for the county. For more information, refer to *Template: BI Publisher Report: United States - State, County and City Geography Codes*
AreaThree	City (or tax district for Pennsylvania) information for the balance value. Supply the geocode for the city, or tax district for Pennsylvania. For more information, refer to *Template: BI Publisher Report: United States - State, County and City Geography Codes*
AreaFour	School district information for the balance value. Supply the geocode for the school district.

Related Topics

- Overview of Balance Initialization
- Steps to Initialize Balances

Example of Loading US Balance Initialization Lines

In this example, you create balance initialization records to initialize an employee's federal and provincial-level balances.

When initializing year-to-date balances, you also initialize any related balances. For example, when initializing Federal Tax Withheld balance, related balances such as Federal Tax Subject and Federal Tax Gross may also need to be initialized, depending on your business needs.

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|LegislativeDataGroupName|BatchName|UploadDate MERGE|
InitializeBalanceBatchHeader|US LDG|Balance Init Group A|2021/02/05
```

In this example, a number of balances are initialized for the same employee.

The employee is identified by their payroll relationship number, employment terms number and assignment number.

Attribute	Value
PayrollRelationshipNumber	8191951
TermNumber	ET8191951
AssignmentNumber	E8191951

The employee's payroll and tax reporting unit must also be specified on every initialize balance line.

Attribute	Value
PayrollName	Vision US Weekly
TaxUnitName	US Tax Reporting Unit

Load these details to initialize the YTD salary balance to 6000:

Attribute	Value
LineSequence	1
BalanceName	Vision US Salary
DimensionName	Assignment Tax Unit Year to Date
AreaOne	
Value	6000

Load these details to initialize the QTD salary balance to 6000

Attribute	Value
LineSequence	2
BalanceName	Vision US Salary
DimensionName	Assignment Tax Unit Quarter to Date
AreaOne	
Value	6000

Load these details to initialize the YTD Federal Income Tax (FIT) Withheld balance

Attribute	Value
LineSequence	3
BalanceName	FIT Withheld
DimensionName	Relationship Tax Unit Year to Date
AreaOne	
Value	954.06

Load these details to initialize the QTD FIT Withheld balance

Attribute	Value
LineSequence	4
BalanceName	FIT Withheld
DimensionName	Relationship Tax Unit Quarter to Date
AreaOne	
Value	954.06

Load these details to initialize the Year-to-Date (YTD) Medicare Employee Withheld balance.

Attribute	Value
LineSequence	5
BalanceName	Medicare Employee Withheld
DimensionName	Relationship Tax Unit Year to Date
AreaOne	

Attribute	Value
Value	87

Load these details to initialize the QTD Medicare Employee Withheld balance.

Attribute	Value
LineSequence	6
BalanceName	Medicare Employee Withheld
DimensionName	Relationship Tax Unit Quarter to Date
AreaOne	
Value	87

Load these details to initialize the YTD SIT Withheld balance.

Attribute	Value
LineSequence	7
BalanceName	SIT Withheld
DimensionName	Relationship Tax Unit, State Year to Date
AreaOne	
Value	264.00

Load these details to initialize the Quarter-to-Date (QTD) SIT Withheld balance.

Attribute	Value
LineSequence	7
BalanceName	SIT Withheld
DimensionName	Relationship Tax Unit, State Quarter to Date
AreaOne	
Value	264.00

Use the InitializeBalanceBatchLine.dat file to initialize the YTD and QTD values for these balances, with the details defined above:

- Salary
- FIT Withheld
- Medicare Employee Withheld

- SIT Withheld

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
AreaOne|Value|UploadDate
MERGE|InitializeBalanceBatchLine|US LDG|Balance Init Group A|1|8191951|ET8191951|E8191951|Vision US Weekly|
US Tax Reporting Unit|Vision US Salary|Assignment Tax Unit Year to Date||6000|
MERGE|InitializeBalanceBatchLine|US LDG|Balance Init Group A|2|8191951|ET8191951|E8191951|Vision US Weekly|
US Tax Reporting Unit|Vision US Salary|Assignment Tax Unit Quarter to Date||6000|
MERGE|InitializeBalanceBatchLine|US LDG|Balance Init Group A|3|8191951|ET8191951|E8191951|Vision US Weekly|
US Tax Reporting Unit|FIT Withheld|Relationship Tax Unit Year to Date||954.06|
MERGE|InitializeBalanceBatchLine|US LDG|Balance Init Group A|4|8191951|ET8191951|E8191951|Vision US Weekly|
US Tax Reporting Unit|FIT Withheld|Relationship Tax Unit Quarter to Date||954.06|
MERGE|InitializeBalanceBatchLine|US LDG|Balance Init Group A|5|8191951|ET8191951|E8191951|Vision US Weekly|
US Tax Reporting Unit|Medicare Employee Withheld|Relationship Tax Unit Year to Date||87|
MERGE|InitializeBalanceBatchLine|US LDG|Balance Init Group A|6|8191951|ET8191951|E8191951|Vision US Weekly|
US Tax Reporting Unit|Medicare Employee Withheld|Relationship Tax Unit Quarter to Date||87|
MERGE|InitializeBalanceBatchLine|US LDG|Balance Init Group A|7|8191951|ET8191951|E8191951|Vision US Weekly|
US Tax Reporting Unit|SIT Withheld|Relationship Tax Unit,State Year to Date|6|264.00|
MERGE|InitializeBalanceBatchLine|US LDG|Balance Init Group A|8|8191951|ET8191951|E8191951|Vision US Weekly|
US Tax Reporting Unit|SIT Withheld|Relationship Tax Unit,State Quarter to Date|6|264.00|
```

Note: When initializing specific balances, you may also need to initialize any related balances. This is simply an example to illustrate the process. For example, when initializing Federal Tax Withheld balance, related balances such as Federal Tax Subject and Federal Tax Gross may also need to be initialized, depending on your business needs.

Related Topics

- Overview of Balance Initialization
- Steps to Initialize Balances
- Overview of Balance Initialization for the US

Example of Loading US Balance Adjustments

This example creates a batch header and group record, with various lines and values.

The following example helps you create a batch header and group record, with various lines and values to adjust the Federal Income Tax (FIT) Withheld and State Income Tax (SIT) Withheld balances for two employees to increase the withheld balances by $10.

Note: When adjusting balances you may also need to adjust any related balances, depending on your business needs. This is simply an example to illustrate the process.

Load the Balance Adjustment Header and Group

Use the BalanceAdjustmentHeader.dat file to create the batch header and group records.

```
METADATA|BalanceAdjustmentHeader|LegislativeDataGroupName|BatchName
MERGE|BalanceAdjustmentHeader|US LDG|FIT and SIT Adjustments

METADATA|BalanceAdjustmentGroup|LegislativeDataGroupName|BatchName|EffectiveDate|PayrollName|
ConsolidationSetName|PrepayFlag|BalanceAdjCostFlag
MERGE|BalanceAdjustmentGroup|US LDG|FIT and SIT Adjustments|2020/04/30|Vision US Weekly|Vision Weekly|N|N
```

Load the Balance Adjustment Lines and Values

The Balance Adjustment Lines files lines specify the element you will supply values for:

Batch Line Sequence	Assignment Number	Element Name	Tax Reporting Unit Name
1	E263769	Federal Income Tax	US Tax Reporting Unit
2	E746169	Federal Income Tax	US Tax Reporting Unit
3	E263769	Work State Income Tax	US Tax Reporting Unit
4	E746169	Work State Income Tax	US Tax Reporting Unit

The Balance Adjustment Values file lines specify the input values for the element named in the line record. The **Entry Value** is the value for the **Input Value Name** of the **Element Name**. For example, in the first line below, the Entry Value of $10 is the value specified for the Tax Calculated input value for the Federal Income Tax element.

For the **Work State Income Tax** element, you must specify the tax calculated amount, as well as the context of the state. That's why there are 2 value records for the 1 line record above for the Work State Income Tax element. In the value record below, where State is the Input Value Name, the Entry Value corresponds to the geography code relating to the value. For example, 5 is the geocode for California (CA).

Batch Line Sequence	Element Name	Input Value Name	Entry Value
1	Federal Income Tax	Tax Calculated	10
2	Federal Income Tax	Tax Calculated	10
3	Work State Income Tax	State	5
3	Work State Income Tax	Tax Calculated	10
4	Work State Income Tax	State	6
4	Work State Income Tax	Tax Calculated	10

Use the BalanceAdjustmentLine.dat file to create the batch line and value records.

```
METADATA|BalanceAdjustmentLine|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|AssignmentNumber|ElementName|TaxReportingUnitName
MERGE|BalanceAdjustmentLine|US LDG|FIT and SIT Adjustments|Vision US Weekly|Vision Weekly|2020/04/30|1|
E263769|Federal Income Tax|US Tax Reporting Unit
MERGE|BalanceAdjustmentLine|US LDG|FIT and SIT Adjustments|Vision US Weekly|Vision Weekly|2020/04/30|2|
E746169|Federal Income Tax|US Tax Reporting Unit
MERGE|BalanceAdjustmentLine|US LDG|FIT and SIT Adjustments|Vision US Weekly|Vision Weekly|2020/04/30|3|
E263769|Work State Income Tax|US Tax Reporting Unit
MERGE|BalanceAdjustmentLine|US LDG|FIT and SIT Adjustments|Vision US Weekly|Vision Weekly|2020/04/30|4|
E746169|Work State Income Tax|US Tax Reporting Unit

METADATA|BalanceAdjustmentValue|LegislativeDataGroupName|BatchName|PayrollName|ConsolidationSetName|
EffectiveDate|BatchLineSequence|InputValueName|ElementName|EntryValue
MERGE|BalanceAdjustmentValue|US LDG|FIT and SIT Adjustments|Vision US Weekly|Vision Weekly|2020/04/30|1|Tax
 Calculated|Federal Income Tax|10
MERGE|BalanceAdjustmentValue|US LDG|FIT and SIT Adjustments|Vision US Weekly|Vision Weekly|2020/04/30|2|Tax
 Calculated|Federal Income Tax|10
MERGE|BalanceAdjustmentValue|US LDG|FIT and SIT Adjustments|Vision US Weekly|Vision Weekly|2020/04/30|3|
State|Work State Income Tax|5
```

```
MERGE|BalanceAdjustmentValue|US LDG|FIT and SIT Adjustments|Vision US Weekly|Vision Weekly|2020/04/30|3|Tax
 Calculated|Work State Income Tax|10
MERGE|BalanceAdjustmentValue|US LDG|FIT and SIT Adjustments|Vision US Weekly|Vision Weekly|2020/04/30|4|
State|Work State Income Tax|6
MERGE|BalanceAdjustmentValue|US LDG|FIT and SIT Adjustments|Vision US Weekly|Vision Weekly|2020/04/30|4|Tax
 Calculated|Work State Income Tax|10
```

Adjust US Balances

In addition to the general guidance provided in the Balance Adjustments section, consider the following when adjusting US balances:

- If an employee has been transferred between states, you must create a separate adjustment line item for each state. Adjustments use the state as a context.
- In order to load contexts for regional levels (state, county, city and school), refer to the Cloud Customer Connect topic *Template: BI Publisher Report: United States - State, County and City Geography Codes* for a report to extract the geocode values.

For additional sample data, supported dimensions and rules, refer to the Balance Adjustment technical brief, located in the US Information Center.

US Information Center

https://support.oracle.com/rs?type=doc&id=2063588.2

US – Payroll tab > Product Documentation > Technical Briefs > Balance Adjustments – Batch.

Involuntary Deductions Card

Overview of Involuntary Deductions for US Employees

This guide describes the guidelines and examples for loading cards and components for processing involuntary deductions in payroll and subsequent post processes.

The Involuntary Deductions card stores information for the processing of involuntary deductions in payroll and subsequent post processes.

Considerations and Prerequisites

A number of prerequisites must be configured prior to loading the Involuntary Deductions card or the card components:

- Involuntary deduction elements
- Element eligibility
- Third-party payees
- Third-party payment methods

Create a card component for each involuntary deduction an employee has, with the appropriate values provided. The component detail and enterable calculation values can vary by involuntary deduction type and state.

The Involuntary Deduction Common Calculation Rules card component is not created by default when the Involuntary Deductions card is initially created through the HCM Data Loader as this component is no longer used.

Involuntary Deductions Card Record Types

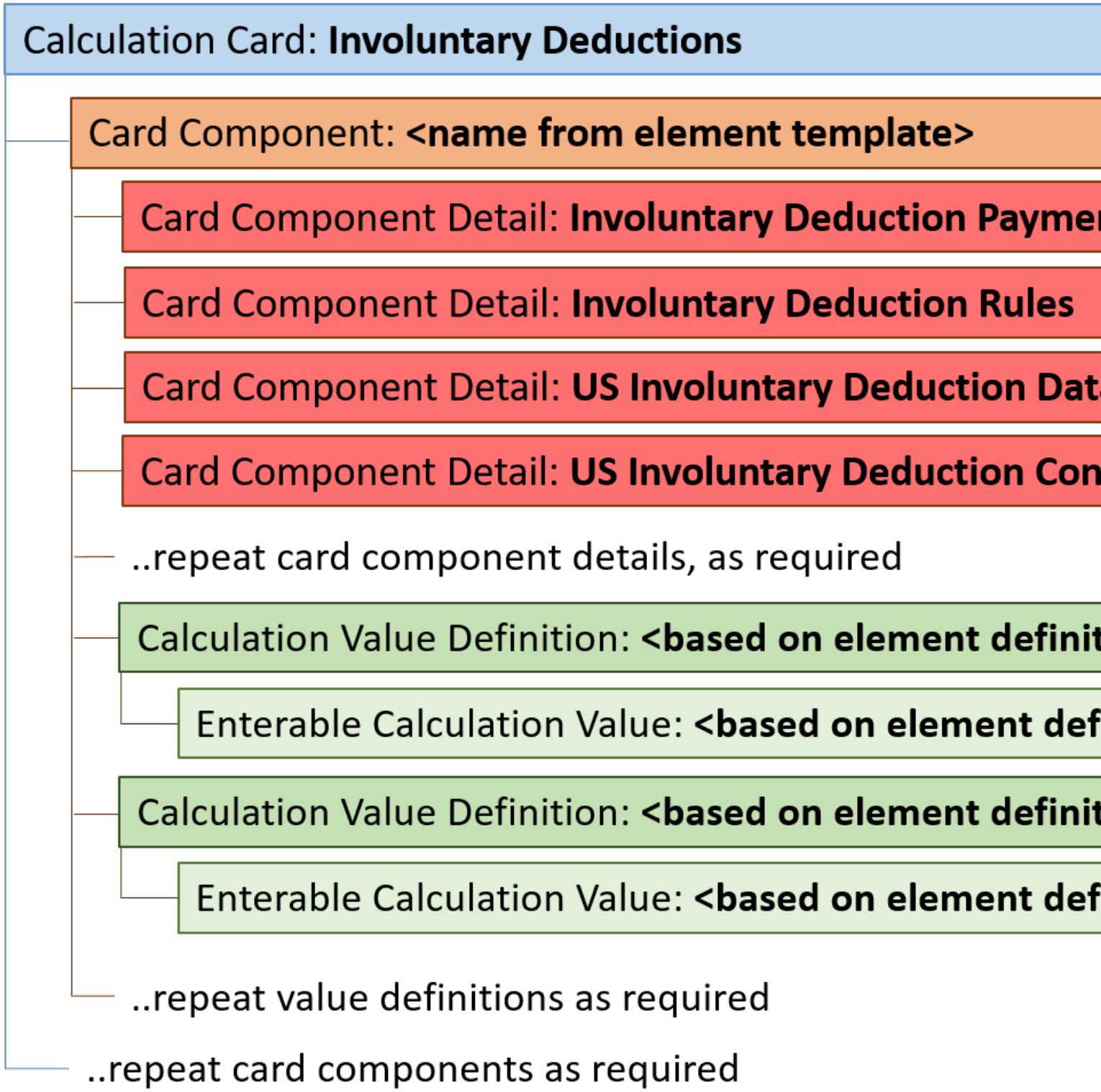
The Involuntary Deduction card is bulk-loaded using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various localization requirements.

The Involuntary Deduction card utilizes the following Calculation Card record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee payroll relationship that it captures information for.	CalculationCard
Card Component	A card component is required for each involuntary deduction. The **Load a Card Component for US Involuntary Deduction** topic describes how to create a card component.	CardComponent
Component Detail	Supplies a component detail record for each flexfield context required by each card component.	ComponentDetail
Calculation Value Definition	Allows the creation of value definitions so that overriding values can be specified on the card component. Details of the specific value definitions are provided in the following sections.	CalculationValueDefinition
Enterable Calculation Value	Specifies an overriding value for each calculation value definition.	EnterableCalculationValue

Involuntary Deduction Calculation Card Hierarchy

The hierarchy of Calculation Card components applicable to Involuntary Deductions is dependent upon the type of involuntary deduction being reported, but generally has this shape:



A separate card component is required for each involuntary deduction type. There is only one Involuntary Deduction card for an employee.

The card component name is defined by the element configuration, as are the value definitions which are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

There are five flexfield contexts available, of which the US Involuntary Deduction Contact Information context is optional. Data for these is loaded using the Component Detail record type. When loading child support orders, when you include these component details, if you exceed the character limit, you may receive a warning that says:

"Data line in file CalculationCard.dat is more than 4000 characters. For reporting purposes the data line will be truncated, but all data will be transferred."

You can ignore this message or remove unnecessary fields from the component detail metadata row.

See these topics that describe how to supply valid file lines for these record types:

- Loading Involuntary Deduction Calculation Cards for US Employees
- Example of Loading Bankruptcy Involuntary Deductions for US Employees
- Example of Loading Child Support, Spousal Support and Alimony Involuntary Deductions for US Employees
- Example of Loading Education Loan and DCIA Involuntary Deductions for US Employees
- Example of Loading Federal Tax Levy Involuntary Deductions for US Employees
- Example of Loading Regional Tax Levy Involuntary Deductions for US Employees
- Example of Loading Garnishment and Creditor Debt Involuntary Deductions for US Employees

Load Involuntary Deduction Calculation Cards for US Employees

Create a Calculation Card record for every employee you are maintaining involuntary deductions for.

Even if you are updating components of an existing calculation card and the calculation card record itself is not being updated, still include the calculation card record to group other related data supplied in the file.

Involuntary Deductions Calculation Card Attributes

The Involuntary Deductions calculation card for employees uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
SourceSystemId	CardSequence, PayrollRelationshipNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Involuntary Deductions calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Involuntary Deductions'

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The start date of the calculation card.
EffectiveEndDate	N/A	The end date of the calculation card.
CardSeqeunce	N/A	Specify '1'. Not required when source keys are used.
PayrollRelationshipNumber	N/A	The number that identifies the employee's payroll relationship.

These attributes are supplied against the CalculationCard file discriminator.

Load a Card Component for US Involuntary Deductions

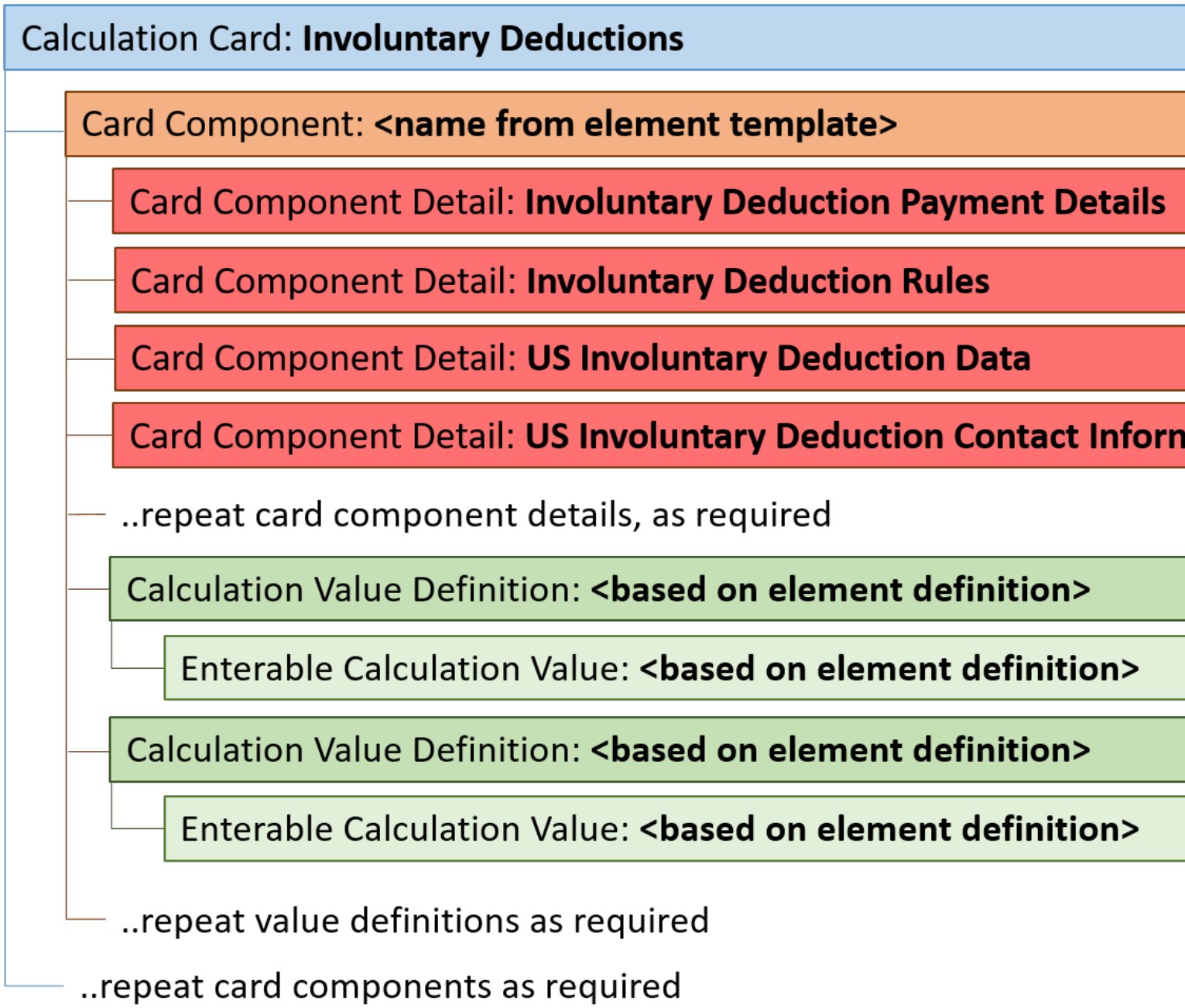
This topic provides general guidance on how to create a Card Component for an Involuntary Deduction.

All Involuntary Deductions are created as individual card components on the same Involuntary Deduction card. An employee will only have one Involuntary Deduction card.

This topic provides general guidance on how to create a Card Component for an Involuntary Deduction. Also review the topic for the specific type of involuntary deduction you are loading.

The general card component hierarchy for an Involuntary:

Deduction:



The card component name is defined by the element configuration, as are the value definitions which are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

The flexfield contexts to load depend on the involuntary deduction type and the element configuration.

Card Component Attributes

The Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the involuntary deductioncard component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent **Involuntary Deductions** calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the **Involuntary Deduction** calculation card. If updating an existing involuntary deductioncard component, the effective start date must be the original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify the name of the element as generated by the element template.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.
Context1	N/A	The value to supply varies depending on the involuntary deduction type. Refer to the example topic of the relevant involuntary deduction type you're loading.
Context2	N/A	The value to supply varies depending on the involuntary deduction type. Refer to the example topic of the relevant involuntary deduction type you're loading.
Subpriority	N/A	An optional value to override the default priority generated.

Component Detail Attributes

The component detail record type enables you to upload data into flexfield segments.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context specific to the Involuntary Deduction type.

Core attributes for Component Details:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent involuntary deduction card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the involuntary deduction card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'INVLN_DEDN_PAYEE_DETAILS'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Involuntary Deduction Component Detail Flexfield Contexts

These four flexfields are applicable to all Involuntary Deduction types:

- Involuntary Deduction Payment Details (INVLN_DEDN_PAYEE_DETAILS)
- Involuntary Deduction Rules (INVLN_DEDN_SUPPORT_DATA)
- US Involuntary Deduction Data (HRX_US_INV_DEDN_DATA)
- US Involuntary Deduction Contact Information (HRX_US_INV_DEDN_CONTACT_INFO_DATA)

Note: The US Involuntary Deduction Contact Information (HRX_US_INV_DEDN_CONTACT_INFO_DATA) flexfield is optional.

For Child Support, Spousal Support and Alimony involuntary deductions, a fifth flexfield context is available:

- US Involuntary Deduction Child Data (HRX_US_INV_DEDN_CHILD_DATA)

You can find the flexfield segment attribute names on the Flexfield Attributes tab of the **View Business Objects** task.

1. Search for the Calculation Card business object and click the object name.
2. Click Component Details in the object hierarchy.
3. Select the Flexfield Attributes tab.
4. Search for each flexfield context in turn to review the list attribute names and the validation set used for attribute validation.

Note: When supplying values to lookup validated flexfield segments, you can provide the lookup meaning using the '_Display' suffixed attribute name, but it is recommended that you supply the lookup code to remove potential translation issues.To supply third party payee details on the Involuntary Deduction Payment Details flexfield context, refer to the Cloud Customer Connect HCM Integrations Resource Sharing Center topic *Template: BI Publisher Report: Third Party Payees*. This report downloads the third-party names and party IDs for the country, legislation and payee type specified.

Flexfield Segment Lookup Validated Attributes

Take a look at these lookup types that are used by the flexfield attributes for the US Involuntary Deductions card:

FLEX:Deduction DeveloperDF	Attribute Name	Lookup Type
HRX_US_INV_DEDN_DATA	_Filing_Status	ORA_HRX_US_INVOL_FILING_STATUS
HRX_US_INV_DEDN_DATA	involDedEarnType	ORA_HRX_US_INVOL_EARN_TYPE
INVLN_DEDN_SUPPORT_DATA	frequency	PAY_PROC_PERIOD_TYPE

Supply the lookup code value to these attributes.

Calculation Value Definition Attributes

Involuntary Deduction card component use value definitions to supply override values.

The value definitions to supply are determined by the element used for the card component. Refer to the Cloud Customer Connect topic *Template: BI Publisher Report: United States - Involuntary Deduction Value Definitions*.

The Calculation Value Definition record type specifies the name of the value definition you are supplying an override value for.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent involuntary deduction card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent involuntary deduction card component. Or, the date the calculation value definition starts, if it's a later date.
DirCardCompDefName	N/A	The definition name of the parent involuntary deduction card component.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed listed in the *Template: BI Publisher Report: United States - Involuntary Deduction Value Definitions* Cloud Customer Connect topic.
ValueType	N/A	The optional type of the value definition, such as Rate or Total amount. Supply a code from the lookup type PAY_DED_OVERRIDE_TYPE.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent involuntary deduction card component and a CalculationCard record for the owning **Involuntary Deductions** card.

Enterable Calculation Value Attributes

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

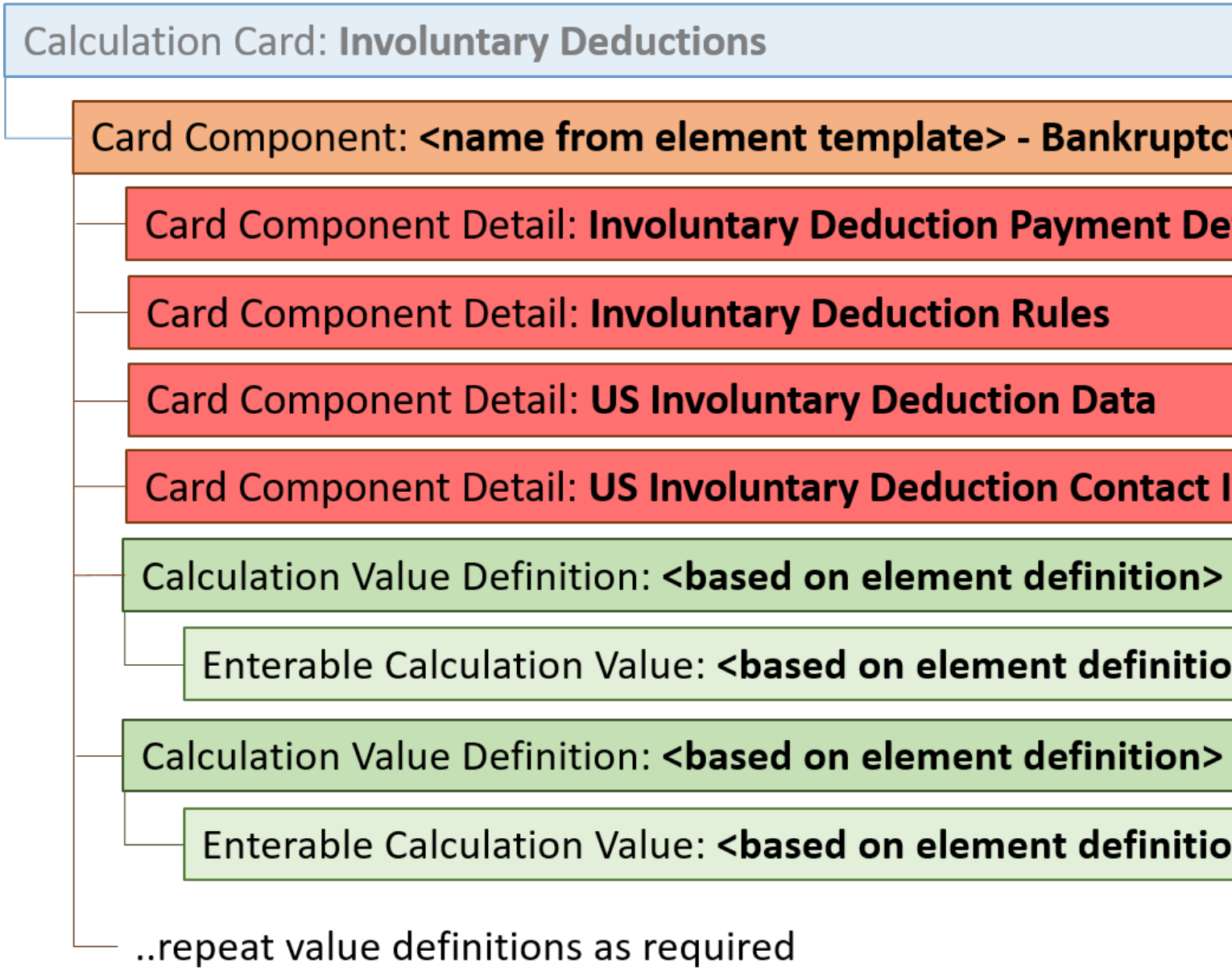
HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Example of Loading Bankruptcy Involuntary Deductions for US Employees

Here's an example of how to load bankruptcy involuntary deductions.

A bankruptcy order is issued as a result of a Federal court procedure that helps individuals get rid of their debts and repay their creditors. An involuntary deduction element using this secondary classification needs to be configured prior to creating the card component.



The card component name is defined by the element configuration, as are the value definitions which are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to the **Load a Card Component for US Involuntary Deductions** for details of each of the record types and attributes you need to supply for an involuntary deduction.

The example file lines in the section below are always supplied together in the same CalculationCard.dat file.

Calculation Card

Always supply the CalculationCard record for the Involuntary Deduction, even if the calculation card already exists for the employee.

Here let's create a new Involuntary Deductions card for Payroll Relationship Number 6485851 starting 03-Jan-2022.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|EffectiveEndDate|CardSequence|PayrollRelationshipNumber
MERGE|CalculationCard|VISION|ID6485851|USA LDG|Involuntary Deductions|2022/01/03|||6485851
```

Card Component

Define the type of Involuntary Deduction using the CardComponent record type.

For the Bankruptcy involuntary deductions you must supply this context attribute:

Attribute Name	Value
Context1	Supply a reference code. This value must be unique by deduction type.

In this example let's create a Bankruptcy card component from 03-Jan-2022. that has a reference code of Bank Order-567.

```
METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|Context1|Context2|Subpriority
MERGE|CardComponent|VISION|ID_BNKRPTCY_6485851|USA LDG|ID6485851|2022/01/03||Vision Bankruptcy|Bank
 Order-567||
```

The parent Involuntary Deduction Calculation Card is identified by the DirCardId(SourceSystemId) attribute which has a value that match the SourceSystemId attribute on the CalculationCard record.

Component Details

Use the ComponentDetail record type to provide flexfield segment values. This example supplies a subset of the flexfield segments available.

```
METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|orderAmtPayee_Display(Deduction Developer DF=INVLN_DEDN_PAYEE_DETAILS)|
receivedDate(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|startDate(Deduction Developer
 DF=INVLN_DEDN_SUPPORT_DATA)|description(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|
issuingAuthorityName(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|remittanceIdentifier(Deduction
 Developer DF=HRX_US_INV_DEDN_DATA)|third2dpartyInvoluntaryDe(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|
documentTrackingNumber(Deduction Developer DF=HRX_US_INV_DEDN_DATA)
MERGE|ComponentDetail|VISION|ID_BNKRPTCY_IDPD_6485851|USA LDG|ID_BNKRPTCY_6485851|2022/01/03||Vision
 Bankruptcy|INVLN_DEDN_PAYEE_DETAILS|INVLN_DEDN_PAYEE_DETAILS|Bankruptcy Trustee||||||
MERGE|ComponentDetail|VISION|ID_BNKRPTCY_IDSD_6485851|USA LDG|ID_BNKRPTCY_6485851|2022/01/03||Vision
 Bankruptcy|INVLN_DEDN_SUPPORT_DATA|INVLN_DEDN_SUPPORT_DATA||2022/01/03|2022/01/03|Bankruptcy Order||||
MERGE|ComponentDetail|VISION|ID_BNKRPTCY_IDD_6485851|USA LDG|ID_BNKRPTCY_6485851|2022/01/03||Vision
 Bankruptcy|HRX_US_INV_DEDN_DATA|HRX_US_INV_DEDN_DATA||||||8889907|1|DocID1235
```

The parent Card Component is identified on the ComponentDetails records by the DirCardCompId(SourceSystemId) attribute, which has a value that matches the SourceSystemId on the CardComponent record.

Value Definitions

Use the CalculationValueDefinition and EnterableValueDefinition record types to provide override values for the value definitions.

In this example, let;s supply the Bankruptcy Order Total Owed Amount value definition with a value of 42000.56.

> **Note:** Refer to the Cloud Customer Connect topic: *US Involuntary Deductions Value Definitions* BI publisher report, to identify all value definitions for the Bankruptcy element.

```
METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|ID_BNKRPTCY_VD1_6485851|USA LDG|ID_BNKRPTCY_6485851|2022/01/03||
Vision Bankruptcy|Bankruptcy Order Total Owed Amount

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|Value1
MERGE|EnterableCalculationValue|VISION|ID_BNKRPTCY_VD1_ECV_6485851|USA LDG|ID_BNKRPTCY_VD1_6485851|
2022/01/03||42000.56
```

The parent Card Component is identified on the CalculationValueDefinitions records by the SourceId(SourceSystemId) attribute which has a value that matches the SourceSystemId on the CardComponent record.

The parent for each Calculation Value Definition is identified on each EnterableCalculationValue record using the ValueDefnId(SourceSystemId) attribute, which has a value that matches the SourceSystemID attribute on the parent CalculationValueDefinition record.

Related Topics

- Overview of Involuntary Deductions for US Employees
- Load Involuntary Deduction Calculation Cards for US Employees
- Load a Card Component for US Involuntary Deductions

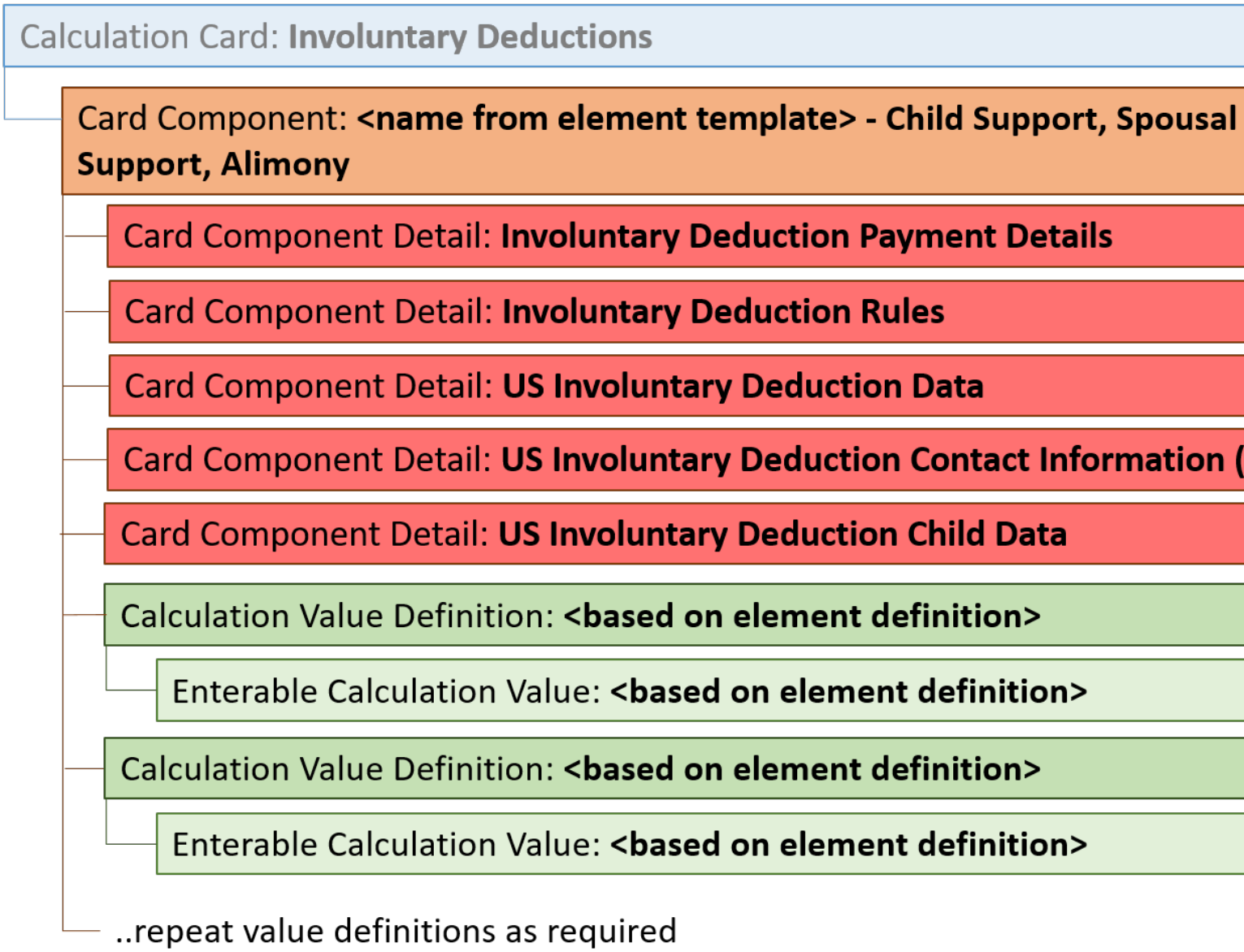
Example of Loading Child Support, Spousal Support and Alimony Involuntary Deductions for US Employees

Child support orders are for payments a noncustodial parent makes as a contribution to the cost of raising their child.

Spousal support orders are for payments for support of an ex-spouse (or a spouse while a divorce is pending) as ordered by the court. Alimony orders are for payments for support made to a divorced person by the former spouse.

Involuntary deduction elements using these secondary classifications need to be configured prior to creating the card components.

Child Support, Spousal Support, Alimony Involuntary Deduction Hierarchy



The card component name is defined by the element configuration, as are the value definitions which are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to the **Loading a Card Component for an Involuntary Deduction** for details of each of the record types and attributes you need to supply for an involuntary deduction.

The example file lines in the section below are always supplied together in the same CalculationCard.dat file.

Calculation Card

Always supply the CalculationCard record for the Involuntary Deduction, even if the calculation card already exists for the employee.

Here, let's create a new Involuntary Deductions card for Payroll Relationship Number 8193987 starting 03-Jan-2022.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|EffectiveEndDate|CardSequence|PayrollRelationshipNumber
```

```
MERGE|CalculationCard|VISION|ID_8193987|USA LDG|Involuntary Deductions|2022/01/03|||8193987
```

Card Component

Define the type of Involuntary Deduction using the CardComponent record type.

For Child Support, Spousal Support and Alimony involuntary deductions you must supply these context attributes:

Attribute Name	Value
Context1	The geocode of the state. Refer to the Cloud Customer Connect topic *Template: BI Publisher Report: United States - State, County and City Geography Codes*.
Context2	Supply a reference code. This value must be unique by deduction type and state.

In this example let's create a Child Support card component from 03-Jan-2022 for California that has a reference code of CA-34567.

```
METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|Context1|Context2|Subpriority
MERGE|CardComponent|VISION|ID_CS_8193987|USA LDG|ID_8193987|2022/01/03||Vision Child Support|5|CA-34567|
```

The parent Involuntary Deduction Calculation Card is identified by the DirCardId(SourceSystemId) attribute which has a value that match the SourceSystemId attribute on the CalculationCard record.

Component Details

Use the ComponentDetail record type to provide flexfield segment values.

In this example let's load the details of a single child. There are additional attributes available if the details of multiple children need to be loaded.

```
METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|
DirInformationCategory|FLEX:Deduction Developer DF|orderAmtPayee_Display(Deduction Developer
 DF=INVLN_DEDN_PAYEE_DETAILS)|receivedDate(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|
startDate(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|description(Deduction Developer
 DF=INVLN_DEDN_SUPPORT_DATA)|issuingAuthorityName(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|
initialFeeTaken(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|issuingJurisdictionName(Deduction Developer
 DF=INVLN_DEDN_SUPPORT_DATA)|_CSE_AGENCY_CASE_IDENTIFIER(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|
remittanceIdentifier(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|_Issuing_State_Display(Deduction
 Developer DF=HRX_US_INV_DEDN_DATA)|_FIPS_Code(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|
_Support_Other_Family(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|_Medical_Support_Indicator(Deduction
 Developer DF=HRX_US_INV_DEDN_DATA)|_PAYMENTS_IN_ARREARS(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|
_Arrears_Overdue_More_Than_12_W(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|_Filing_Status(Deduction
 Developer DF=HRX_US_INV_DEDN_DATA)|third2dpartyInvoluntaryDe(Deduction Developer
 DF=HRX_US_INV_DEDN_DATA)|documentTrackingNumber(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|
involDedEarnType(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|_Name_of_Child(Deduction Developer
 DF=HRX_US_INV_DEDN_CHILD_DATA)|_Date_of_Birth_of_Child(Deduction Developer DF=HRX_US_INV_DEDN_CHILD_DATA)|
lastNameOfChild1(Deduction Developer DF=HRX_US_INV_DEDN_CHILD_DATA)|firstNameOfChild1(Deduction Developer
 DF=HRX_US_INV_DEDN_CHILD_DATA)|middleNameOfChild1(Deduction Developer DF=HRX_US_INV_DEDN_CHILD_DATA)

MERGE|ComponentDetail|VISION|ID_CD_IDPD_8193987|USA LDG|ID_CS_8193987|2022/01/03||Vision Child Support|
INVLN_DEDN_PAYEE_DETAILS|INVLN_DEDN_PAYEE_DETAILS|Child Support SDU|||||||||||||||||||||||
MERGE|ComponentDetail|VISION|ID_CD_IDSD_8193987|USA LDG|ID_CS_8193987|2022/01/03||Vision Child Support|
INVLN_DEDN_SUPPORT_DATA|INVLN_DEDN_SUPPORT_DATA||2022/01/03|2022/01/10|California Child Support Order|
California SDU|Y|California|||||||||||||||
MERGE|ComponentDetail|VISION||ID_CD_IDD_8193987|USA LDG|ID_CS_8193987|2022/01/03||Vision Child Support|
HRX_US_INV_DEDN_DATA|HRX_US_INV_DEDN_DATA|||||||CSE-12345|8883346|CA|06000|Y|Y|Y|N||1|DocID12345||||||
```

```
MERGE|ComponentDetail|VISION|ID_CD_IDCD_955160008193987|USA LDG|ID_CS_8193987|2022/01/03||Vision Child
 Support|HRX_US_INV_DEDN_CHILD_DATA|HRX_US_INV_DEDN_CHILD_DATA|||||||||||||||||||||2015/06/03|Child1 Last
 Name|Child1 First Name|Child1 Middle Name
```

The parent Card Component is identified on the ComponentDetails records by the DirCardCompId(SourceSystemId) attribute, which has a value that matches the SourceSystemId on the CardComponent record.

Value Definitions

Use the CalculationValueDefinition and EnterableValueDefinition record types to provide override values for the value definitions.

Here the following value definitions are overridden:

Value Definition	Value
California Child Support Total Withholding Amount	230.77
Proration Child Support Arrears Amount	130.77
Proration Child Support Current Amount	100.00

Note: Refer to the Cloud Customer Connect topic *Template: BI Publisher Report: United States - Involuntary Deduction Value Definitions* to identify all value definitions for the Child Support, Spousal Support or Alimony element.

```
METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|ID_CS_VD1_8193987|USA LDG|ID_CS_8193987|2022/01/03||Vision Child
 Support|California Child Support Total Withholding Amount
MERGE|CalculationValueDefinition|VISION|ID_CS_VD2_8193987|USA LDG|ID_CS_8193987|2022/01/03||Vision Child
 Support|Proration Child Support Arrears Amount
MERGE|CalculationValueDefinition|VISION|ID_CS_VD3_8193987|USA LDG|ID_CS_8193987|2022/01/03||Vision Child
 Support|Proration Child Support Current Amount

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|Value1
MERGE|EnterableCalculationValue|VISION|ID_CS_VD1_ECV_8193987|USA LDG|ID_CS_VD1_8193987|2022/01/03||230.77
MERGE|EnterableCalculationValue|VISION|ID_CS_VD2_ECV_8193987|USA LDG|ID_CS_VD2_8193987|2022/01/03||130.77
MERGE|EnterableCalculationValue|VISION|ID_CS_VD3_ECV_8193987|USA LDG|ID_CS_VD3_8193987|2022/01/03||100
```

The parent Card Component is identified on the CalculationValueDefinitions records by the SourceId(SourceSystemId) attribute which has a value that matches the SourceSystemId on the CardComponent record.

The parent for each Calculation Value Definition is identified on each EnterableCalculationValue record using the ValueDefnId(SourceSystemId) attribute, which has a value that matches the SourceSystemID attribute on the parent CalculationValueDefinition record.

Related Topics

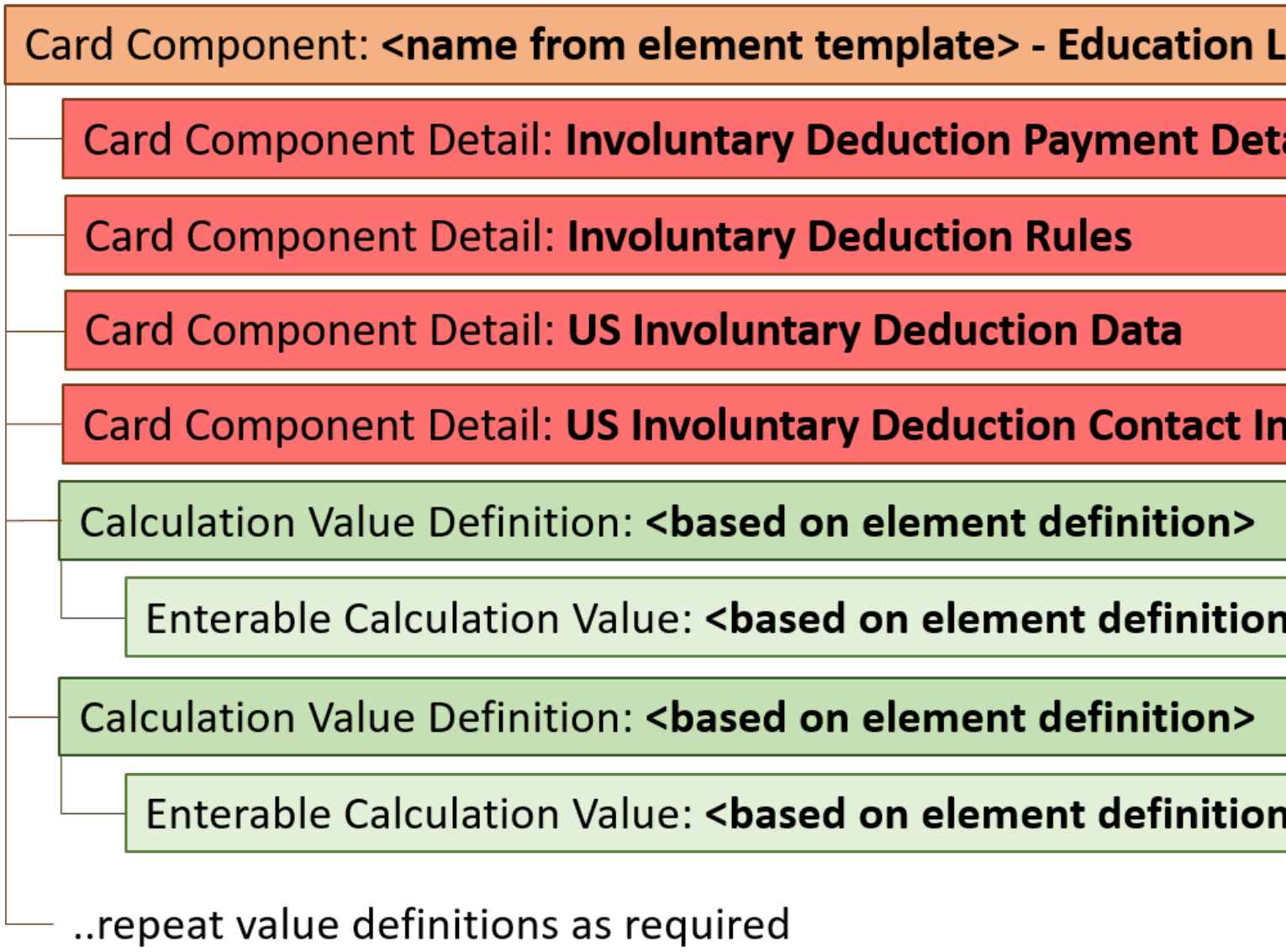
- Overview of Involuntary Deductions for US Employees
- Load Involuntary Deduction Calculation Cards for US Employees
- Load a Card Component for US Involuntary Deductions

Example of Loading Education Loan and DCIA Involuntary Deductions for US Employees

Education Loan orders are for delinquent loans for education granted under the Federal Direct Loan Program or Federal Family Education Loan Program.

Debt Collection Improvement Act (DCIA) orders are issued by Federal agencies for non-tax debts owed to the U.S. government.

Involuntary deduction elements using these secondary classifications need to be configured prior to creating the card components.



The card component name is defined by the element configuration, as are the value definitions which are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to the **Load a Card Component for US Involuntary Deductions** for details of each of the record types and attributes you need to supply for an involuntary deduction.

The example file lines in the section below are always supplied together in the same CalculationCard.dat file.

Calculation Card

Always supply the CalculationCard record for the Involuntary Deduction, even if the calculation card already exists for the employee.

Here let's create a new Involuntary Deductions card for Payroll Relationship Number 6485851 starting 03-Jan-2022.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|EffectiveEndDate|CardSequence|PayrollRelationshipNumber
MERGE|CalculationCard|VISION|ID_6485851|USA LDG|Involuntary Deductions|2022/01/03|||6485851
```

Card Component

Define the type of Involuntary Deduction using the CardComponent record type.

For the Education Loan and DCIA involuntary deductions you must supply this context attribute:

Attribute Name	Value
Context1	Supply a reference code. This value must be unique by deduction type.

In this example let's create a Bankruptcy card component from 03-Jan-2022 that has a reference code of EDUL 32688

```
METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|Context1|Context2|Subpriority
MERGE|CardComponent|VISION|ID_EDU_LOAN_6485851|USA LDG|ID_6485851|2022/01/03||Vision Education Loan|
EDUL-32688||
```

The parent Involuntary Deduction Calculation Card is identified by the DirCardId(SourceSystemId) attribute which has a value that match the SourceSystemId attribute on the CalculationCard record.

Component Details

Use the ComponentDetail record type to provide flexfield segment values. This example supplies a subset of the flexfield segments available.

```
METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|orderAmtPayee_Display(Deduction Developer DF=INVLN_DEDN_PAYEE_DETAILS)|
receivedDate(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|startDate(Deduction Developer
 DF=INVLN_DEDN_SUPPORT_DATA)|description(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|
issuingAuthorityName(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|initialFeeTaken(Deduction Developer
 DF=INVLN_DEDN_SUPPORT_DATA)|issuingJurisdictionName(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|
remittanceIdentifier(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|third2dpartyInvoluntaryDe(Deduction
 Developer DF=HRX_US_INV_DEDN_DATA)|documentTrackingNumber(Deduction Developer DF=HRX_US_INV_DEDN_DATA)
MERGE|ComponentDetail|VISION|ID_EDU_LOAN_IDPD_6485851|USA LDG|ID_EDU_LOAN_6485851|2022/01/03||Vision
 Education Loan|INVLN_DEDN_PAYEE_DETAILS|INVLN_DEDN_PAYEE_DETAILS|Education Loan Processor|||||||||
MERGE|ComponentDetail|VISION|ID_EDU_LOAN_IDSD_6485851|USA LDG|ID_EDU_LOAN_6485851|2022/01/03||Vision
 Education Loan|INVLN_DEDN_SUPPORT_DATA|INVLN_DEDN_SUPPORT_DATA||2022/01/03||Student Loan||||||
```

```
MERGE|ComponentDetail|VISION|ID_EDU_LOAN_IDD_6485851|USA LDG|ID_EDU_LOAN_6485851|2022/01/03||Vision
 Education Loan|HRX_US_INV_DEDN_DATA|HRX_US_INV_DEDN_DATA|||||||6489322|1|DocID1236
```

The parent Card Component is identified on the ComponentDetails records by the DirCardCompId(SourceSystemId) attribute, which has a value that matches the SourceSystemId on the CardComponent record.

Value Definitions

Use the CalculationValueDefinition and EnterableValueDefinition record types to provide override values for the value definitions.

In this example the following value definitions are overridden:

Value Definition	Value
Educational Loan Order Percentage	.15
Educational Loan Total Owed Amount	12344.56

Note: Refer to the Cloud Customer Connect topic: *US Involuntary Deductions Value Definitions* BI Publisher report, to identify all value definitions for the Education Loan or DCIA element.

```
METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|ID_EDU_LOAN_VD1_6485851|USA LDG|ID_EDU_LOAN_6485851|2022/01/03||
Vision Education Loan|Educational Loan Order Percentage
MERGE|CalculationValueDefinition|VISION|ID_EDU_LOAN_VD2_6485851|USA LDG|ID_EDU_LOAN_6485851|2022/01/03||
Vision Education Loan|Educational Loan Total Owed Amount

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|Value1
MERGE|EnterableCalculationValue|VISION|ID_EDU_LOAN_VD1_ECV_6485851|USA LDG|ID_EDU_LOAN_VD1_6485851|
2022/01/03||0.15
MERGE|EnterableCalculationValue|VISION|ID_EDU_LOAN_VD2_ECV_6485851|USA LDG|ID_EDU_LOAN_VD2_6485851|
2022/01/03||12344.56
```

The parent Card Component is identified on the CalculationValueDefinitions records by the SourceId(SourceSystemId) attribute which has a value that matches the SourceSystemId on the CardComponent record.

The parent for each Calculation Value Definition is identified on each EnterableCalculationValue record using the ValueDefnId(SourceSystemId) attribute, which has a value that matches the SourceSystemID attribute on the parent CalculationValueDefinition record.

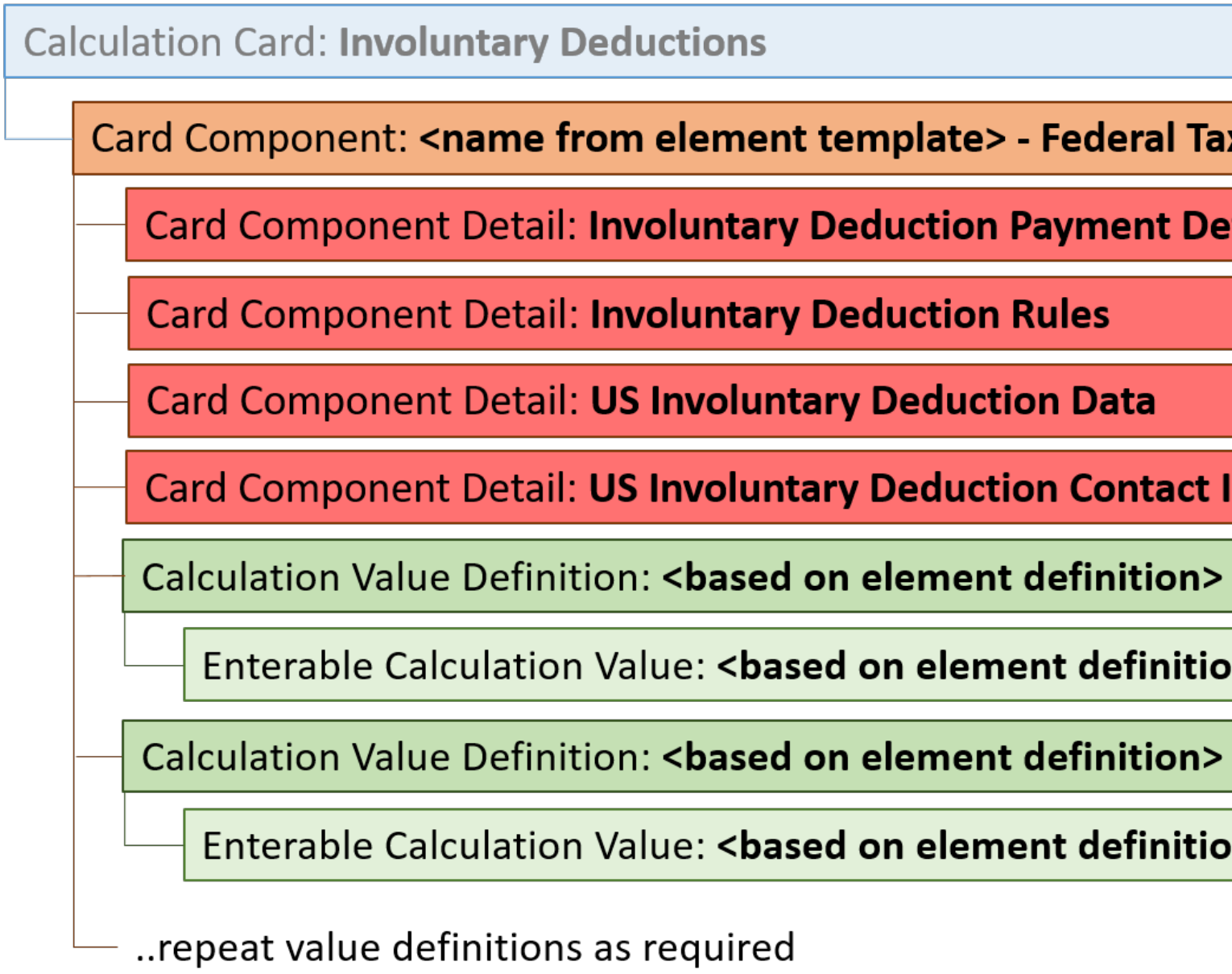
Related Topics

- Overview of Involuntary Deductions for US Employees
- Load Involuntary Deduction Calculation Cards for US Employees
- Load a Card Component for US Involuntary Deductions

Example of Loading Federal Tax Levy Involuntary Deductions for US Employees

A tax levy order is an administrative action by the Internal Revenue Service (IRS) to seize property to satisfy a tax liability.

An involuntary deduction element using this secondary classification needs to be configured prior to creating the card component.



The card component name is defined by the element configuration, as are the value definitions which are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to the **Loading a Card Component for an Involuntary Deduction** for details of each of the record types and attributes you need to supply for an involuntary deduction.

The example file lines in the section below are always supplied together in the same CalculationCard.dat file.

Calculation Card

Always supply the CalculationCard record for the Involuntary Deduction, even if the calculation card already exists for the employee.

Here let's create a new Involuntary Deductions card for Payroll Relationship Number 6485851 starting 03-Jan-2022.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|EffectiveEndDate|CardSequence|PayrollRelationshipNumber
MERGE|CalculationCard|VISION|ID_6485851|USA LDG|Involuntary Deductions|2022/01/03|||6485851
```

Card Component

Define the type of Involuntary Deduction using the CardComponent record type.

For Federal Tax Levy involuntary deductions you must supply this context attribute:

Attribute Name	Value
Context1	Supply a reference code. This value must be unique by deduction type.

In this example, let's create a Federal Tax Levy card component from 01-Jan-2018 that has a reference code of IRS-34586

```
METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|Context1|Context2|Subpriority
MERGE|CardComponent|VISION|ID_TAX_LEVY_6485851|USA LDG|ID_6485851|2022/01/03||Vision Federal Tax Levy|
IRS-34586||
```

The parent Involuntary Deduction Calculation Card is identified by the DirCardId(SourceSystemId) attribute which has a value that matches the SourceSystemId attribute on the CalculationCard record.

Component Details

Use the ComponentDetail record type to provide flexfield segment values. This example supplies a subset of the flexfield segments available.

```
METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|
DirInformationCategory|FLEX:Deduction Developer DF|orderAmtPayee_Display(Deduction Developer
 DF=INVLN_DEDN_PAYEE_DETAILS)|receivedDate(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|
startDate(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|description(Deduction Developer
 DF=INVLN_DEDN_SUPPORT_DATA)|remittanceIdentifier(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|
_Filing_Status_Display(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|_STATEMENT_OF_EXEMPTIONS_REC(Deduction
 Developer DF=HRX_US_INV_DEDN_DATA)|third2dpartyInvoluntaryDe(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|
involDedEarnType(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|documentTrackingNumber(Deduction Developer
 DF=HRX_US_INV_DEDN_DATA)|contactType(Deduction Developer DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|
firstName(Deduction Developer DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|middleName(Deduction Developer
 DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|lastName(Deduction Developer DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|
addressLine1(Deduction Developer DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|addressLine2(Deduction Developer
 DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|city(Deduction Developer DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|
```

```
county(Deduction Developer DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|postalCode(Deduction Developer
 DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|state(Deduction Developer DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|
country(Deduction Developer DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|workPhone(Deduction
 Developer DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA) |mobilePhone(Deduction Developer
 DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|otherPhone(Deduction Developer DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|
emailAddress(Deduction Developer DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)|comments(Deduction Developer
 DF=HRX_US_INV_DEDN_CONTACT_INFO_DATA)
MERGE|ComponentDetail|VISION|ID_TAX_LEVY_IDPD_6485851|USA LDG|ID_TAX_LEVY_6485851|2022/01/03||Vision Federal
 Tax Levy|INVLN_DEDN_PAYEE_DETAILS|INVLN_DEDN_PAYEE_DETAILS|IRS|||||||||||||||||||||||||
MERGE|ComponentDetail|VISION|ID_TAX_LEVY_IDSD_6485851|USA LDG|ID_TAX_LEVY_6485851|2022/01/03||
Vision Federal Tax Levy|INVLN_DEDN_SUPPORT_DATA|INVLN_DEDN_SUPPORT_DATA||2022/01/03||Federal Tax
 Levy|||||||||||||||||||||||
MERGE|ComponentDetail|VISION|ID_TAX_LEVY_IDD_6485851|USA LDG|ID_TAX_LEVY_6485851|2022/01/03||Vision
 Federal Tax Levy|HRX_US_INV_DEDN_DATA|HRX_US_INV_DEDN_DATA|||||8883346|Single|2022/01/03|1||
DocID1237|||||||||||||||||
MERGE|ComponentDetail|VISION|ID_TAX_LEVY_IDC_6485851|USA LDG|ID_TAX_LEVY_6485851|2022/01/03||Vision
 Federal Tax Levy|HRX_US_INV_DEDN_CONTACT_INFO_DATA|HRX_US_INV_DEDN_CONTACT_INFO_DATA||||||||||
Attorney|John|Anthony|Doe|100 Main Street|Suite 200|Atlanta|Fulton|30202|GA|United States||770-555-1212||
john.doe@legal.com|
```

The parent Card Component is identified on the ComponentDetails records by the DirCardCompId(SourceSystemId) attribute, which has a value that matches the SourceSystemId on the CardComponent record.

Value Definitions

Use the CalculationValueDefinition and EnterableValueDefinition record types to provide override values for the value definitions.

In this example the following value definitions are overridden:

Value Definition	Value
Tax Levy Total Owed Amount	5566.77
Tax Levy Total Allowances	2
Tax Levy Deductions At Time of Writ Pretax Amount	100
Tax Levy Deductions At Time of Writ Deferred Compensation Rate	.05

Note: Refer to the Cloud Customer Connect topic: *US Involuntary Deductions Value Definitions* BI Publisher report, to identify all value definitions for the Federal Tax Levy element.

```
METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|ID_TAX_LEVY_VD1_6485851|USA LDG|ID_TAX_LEVY_6485851|2022/01/03||
Vision Federal Tax Levy|Tax Levy Total Owed Amount
MERGE|CalculationValueDefinition|VISION|ID_TAX_LEVY_VD2_6485851|USA LDG|ID_TAX_LEVY_6485851|2022/01/03||
Vision Federal Tax Levy|Tax Levy Total Allowances
MERGE|CalculationValueDefinition|VISION|ID_TAX_LEVY_VD3_6485851|USA LDG|ID_TAX_LEVY_6485851|2022/01/03||
Vision Federal Tax Levy|Tax Levy Deductions At Time of Writ Pretax Amount
MERGE|CalculationValueDefinition|VISION|ID_TAX_LEVY_VD4_6485851|USA LDG|ID_TAX_LEVY_6485851|2022/01/03||
Vision Federal Tax Levy|Tax Levy Deductions At Time of Writ Deferred Compensation Rate

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|Value1|
MERGE|EnterableCalculationValue|VISION|ID_TAX_LEVY_VD1_ECV_6485851|USA LDG|ID_TAX_LEVY_VD1_6485851|
2022/01/03||5566.77
```

```
MERGE|EnterableCalculationValue|VISION|ID_TAX_LEVY_VD2_ECV_6485851|USA LDG|ID_TAX_LEVY_VD2_6485851|
2022/01/03||2
MERGE|EnterableCalculationValue|VISION|ID_TAX_LEVY_VD3_ECV_6485851|USA LDG|ID_TAX_LEVY_VD2_6485851|
2022/01/03||100
MERGE|EnterableCalculationValue|VISION|ID_TAX_LEVY_VD3_ECV_6485851|USA LDG|ID_TAX_LEVY_VD2_6485851|
2022/01/03||0.05
```

The parent Card Component is identified on the CalculationValueDefinitions records by the SourceId(SourceSystemId) attribute which has a value that matches the SourceSystemId on the CardComponent record.

The parent for each Calculation Value Definition is identified on each EnterableCalculationValue record using the ValueDefnId(SourceSystemId) attribute, which has a value that matches the SourceSystemID attribute on the parent CalculationValueDefinition record.

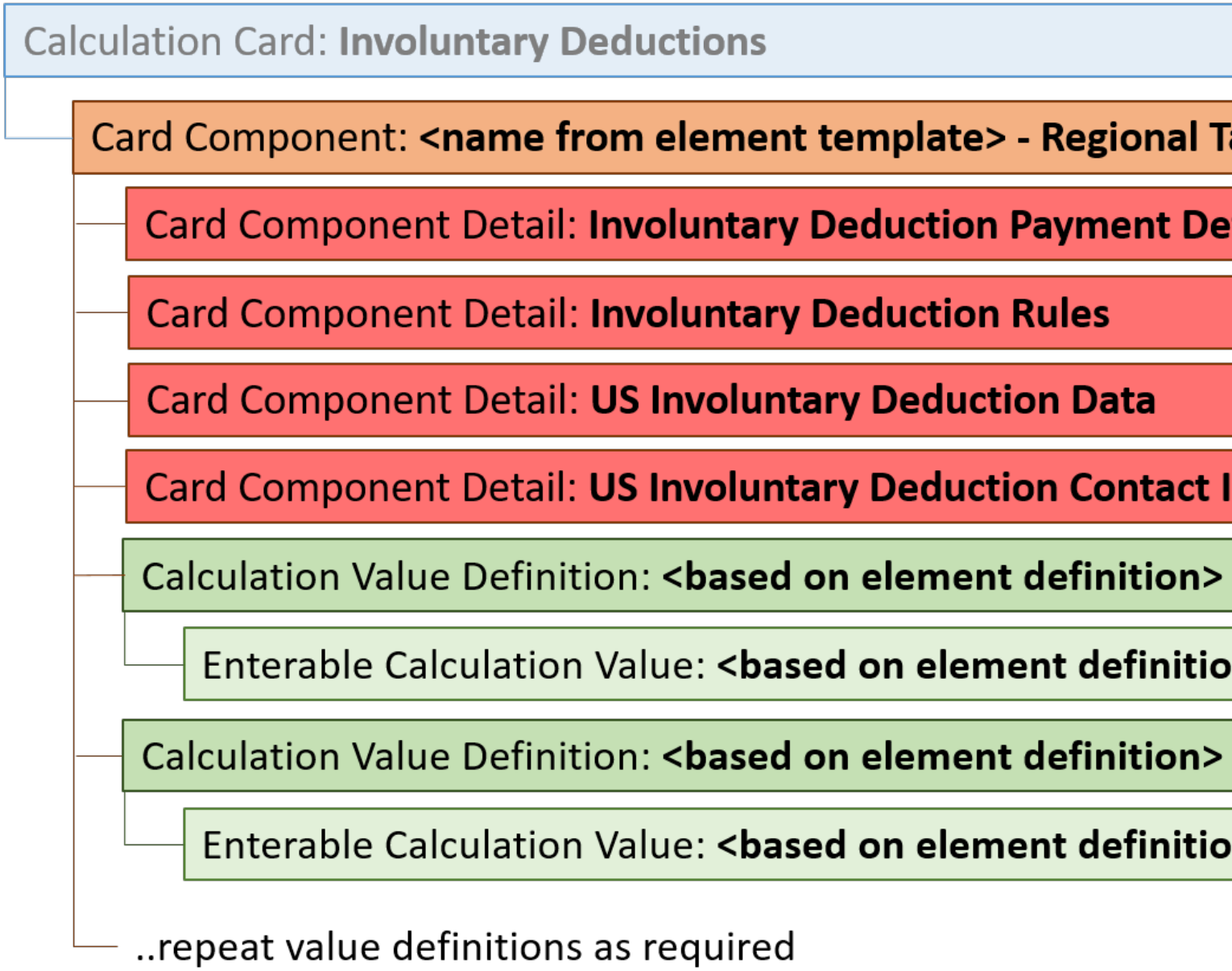
Related Topics

- Overview of Involuntary Deductions for US Employees
- Load Involuntary Deduction Calculation Cards for US Employees
- Load a Card Component for US Involuntary Deductions

Example of Loading Regional Tax Levy Involuntary Deductions for US Employees

Regional Tax Levy orders are issued for legal seizure of taxpayers' assets to satisfy back taxes owed.

An involuntary deduction element using this secondary classification needs to be configured prior to creating the card component.



The card component name is defined by the element configuration, as are the value definitions which are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to the Loading a Card Component for an Involuntary Deduction for details of each of the record types and attributes you need to supply for an involuntary deduction.

The example file lines in the section below are always supplied together in the same CalculationCard.dat file.

Calculation Card

Always supply the CalculationCard record for the Involuntary Deduction, even if the calculation card already exists for the employee.

Here let's create a new Involuntary Deductions card for Payroll Relationship Number 6485851 starting 03-Jan-2022.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|EffectiveEndDate|CardSequence|PayrollRelationshipNumber
MERGE|CalculationCard|VISION|ID_6485851|USA LDG|Involuntary Deductions|2022/01/03|||6485851
```

Card Component

Define the type of Involuntary Deduction using the CardComponent record type.

For Regional Tax Levy involuntary deductions you must supply these context attributes:

Attribute Name	Value
Context1	The geocode of the state. Refer to the Cloud Customer Connect topic *Template: BI Publisher Report: United States - State, County and City Geography Codes*.
Context2	Supply a reference code. This value must be unique by deduction type and state.

In this example, let's create a Regional Tax Levy card component from 03-Jan-2022 that has a reference code of RGL-003.

```
METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|Context1|Context2|Subpriority
MERGE|CardComponent|VISION|ID_RTL_6485851|USA LDG|ID_6485851|2022/01/03||Vision Regional Tax Levy|5|RGL-003|
```

The parent Involuntary Deduction Calculation Card is identified by the DirCardId(SourceSystemId) attribute which has a value that match the SourceSystemId attribute on the CalculationCard record.

Component Details

Use the ComponentDetail record type to provide flexfield segment values. This example supplies a subset of the flexfield segments available.

```
METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|orderAmtPayee_Display(Deduction Developer DF=INVLN_DEDN_PAYEE_DETAILS)|
processFeePayee_Display(Deduction Developer DF=INVLN_DEDN_PAYEE_DETAILS)|receivedDate(Deduction Developer
 DF=INVLN_DEDN_SUPPORT_DATA)|startDate(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|description(Deduction
 Developer DF=INVLN_DEDN_SUPPORT_DATA)|issuingAuthorityName(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|
initialFeeTaken(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|issuingJurisdictionName(Deduction
 Developer DF=INVLN_DEDN_SUPPORT_DATA)|remittanceIdentifier(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|
_Issuing_State_Display(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|_Filing_Status(Deduction Developer
 DF=HRX_US_INV_DEDN_DATA)|third2dpartyInvoluntaryDe(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|
documentTrackingNumber(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|involDedEarnType(Deduction Developer
 DF=HRX_US_INV_DEDN_DATA)
MERGE|ComponentDetail|VISION|ID_RTL_IDPD_6485851|USA LDG|ID_RTL_6485851|2022/01/03||VisionRegional Tax
 Levy|INVLN_DEDN_PAYEE_DETAILS|INVLN_DEDN_PAYEE_DETAILS|Dept of Revenue|Employer Payee for Processing
 Fee|||||||||||
MERGE|ComponentDetail|VISION|ID_RTL_IDSD_6485851|USA LDG|ID_RTL_6485851|2022/01/03||Regional Tax
 Levy|INVLN_DEDN_SUPPORT_DATA|INVLN_DEDN_SUPPORT_DATA|||2022/01/03|2022/01/10|Regional Tax Levy||Y|
California||||||
MERGE|ComponentDetail|VISION|ID_RTL_IDD_6485851|USA LDG|ID_RTL_6485851|2022/01/03||Vision Regional Tax Levy|
HRX_US_INV_DEDN_DATA|HRX_US_INV_DEDN_DATA|||||||||8883346|CA||1|DocID12348|
```

The parent Card Component is identified on the ComponentDetails records by the DirCardCompId(SourceSystemId) attribute, which has a value that matches the SourceSystemId on the CardComponent record.

Value Definitions

Use the CalculationValueDefinition and EnterableValueDefinition record types to provide override values for the value definitions.

In this example the following value definitions are overridden:

Value Definition	Value
California Regional Tax Levy Order Amount	400
California Regional Tax Levy Total Owed Amount	2220.20

Note: Use the *Template: BI Publisher Report: United States - Involuntary Deduction Value Definitions* report to identify all value definitions for the Regional Tax Levy element.

```
METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|ID_RTL_VD1_6485851|USA LDG|ID_RTL_6485851|2022/01/03||Vision
 Regional Tax Levy|California Regional Tax Levy Order Amount
MERGE|CalculationValueDefinition|VISION|ID_RTL_VD2_6485851|USA LDG|ID_RTL_6485851|2022/01/03||Vision
 Regional Tax Levy|California Regional Tax Levy Total Owed Amount

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|Value1
MERGE|EnterableCalculationValue|VISION|ID_RTL_VD1_ECV_6485851|USA LDG|ID_RTL_VD1_6485851|2022/01/03||400.3
MERGE|EnterableCalculationValue|VISION|ID_RTL_VD2_ECV_6485851|USA LDG|ID_RTL_VD2_6485851|2022/01/03||2220.2
```

The parent Card Component is identified on the CalculationValueDefinitions records by the SourceId(SourceSystemId) attribute which has a value that matches the SourceSystemId on the CardComponent record.

The parent for each Calculation Value Definition is identified on each EnterableCalculationValue record using the ValueDefnId(SourceSystemId) attribute, which has a value that matches the SourceSystemID attribute on the parent CalculationValueDefinition record.

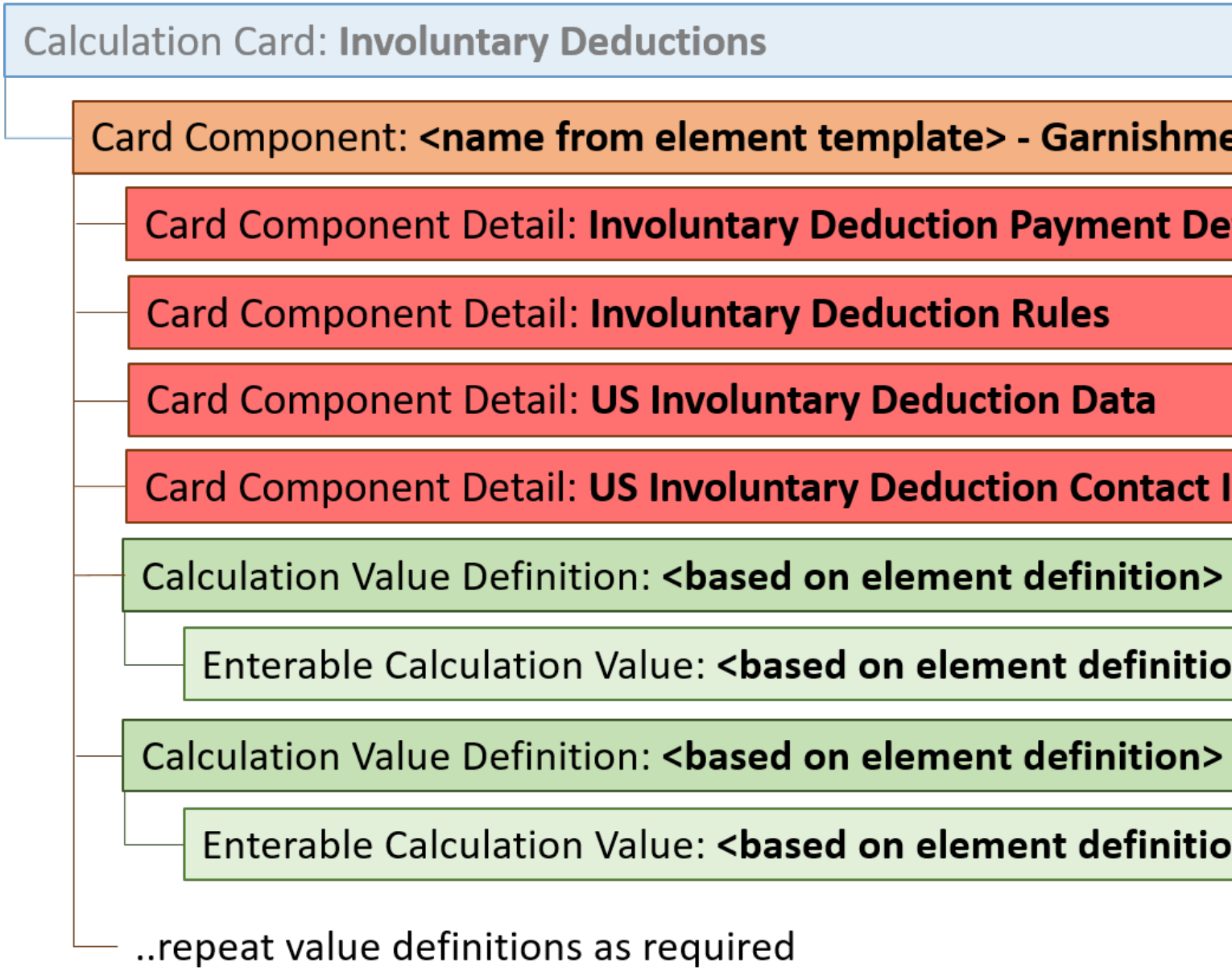
Related Topics

- Overview of Involuntary Deductions for US Employees
- Load Involuntary Deduction Calculation Cards for US Employees
- Load a Card Component for US Involuntary Deductions

Example of Loading Garnishment and Creditor Debt Involuntary Deductions for US Employees

Wage garnishment or creditor debt occurs when an employer is required to withhold the earnings of an employee for the payment of a debt in accordance with a court order or other legal or equitable procedure.

An involuntary deduction element using this secondary classification needs to be configured prior to creating the card component.



The card component name is defined by the element configuration, as are the value definitions which are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

Refer to the **Loading a Card Component for an Involuntary Deduction** for details of each of the record types and attributes you need to supply for an involuntary deduction.

The example file lines in the section below are always supplied together in the same CalculationCard.dat file.

Calculation Card

Always supply the CalculationCard record for the Involuntary Deduction, even if the calculation card already exists for the employee.

Here, let's create a new Involuntary Deductions card for Payroll Relationship Number 6485851 starting 03-Jan-2022.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|EffectiveEndDate|CardSequence|PayrollRelationshipNumber
MERGE|CalculationCard|VISION|ID_6485851|USA LDG|Involuntary Deductions|2022/01/03|||6485851
```

Card Component

Define the type of Involuntary Deduction using the CardComponent record type.

For Garnishment and Creditor Debt involuntary deductions you must supply this context attribute:

Attribute Name	Value
Context1	The geocode of the state. Refer to the Cloud Customer Connect topic *Template: BI Publisher Report: United States - State, County and City Geography Codes*.
Context2	Supply a reference code. This value must be unique by deduction type and state.

In this example, let's create a Garnishment card component from 03-Jan-2022 for the state of Tennessee that has a reference code of Garn-32688.

```
METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|Context1|Context2|Subpriority
MERGE|CardComponent|VISION|ID_GARN_6485851|USA LDG|ID_6485851|2022/01/03||Vision Creditor Debt|43|
Garn-32688|
```

The parent Involuntary Deduction Calculation Card is identified by the DirCardId(SourceSystemId) attribute which has a value that match the SourceSystemId attribute on the CalculationCard record.

Component Details

Use the ComponentDetail record type to provide flexfield segment values. This example supplies a subset of the flexfield segments available.

```
METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|orderAmtPayee_Display(Deduction Developer DF=INVLN_DEDN_PAYEE_DETAILS)|
processFeePayee_Display(Deduction Developer DF=INVLN_DEDN_PAYEE_DETAILS)|receivedDate(Deduction Developer
 DF=INVLN_DEDN_SUPPORT_DATA)|startDate(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|description(Deduction
 Developer DF=INVLN_DEDN_SUPPORT_DATA)|issuingAuthorityName(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|
initialFeeTaken(Deduction Developer DF=INVLN_DEDN_SUPPORT_DATA)|issuingJurisdictionName(Deduction
 Developer DF=INVLN_DEDN_SUPPORT_DATA)|remittanceIdentifier(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|
third2dpartyInvoluntaryDe(Deduction Developer DF=HRX_US_INV_DEDN_DATA)|documentTrackingNumber(Deduction
 Developer DF=HRX_US_INV_DEDN_DATA)
MERGE|ComponentDetail|VISION|ID_GARN_IDPD_6485851|USA LDG|ID_GARN_6485851|2022/01/03||Vision Creditor
 Debt|INVLN_DEDN_PAYEE_DETAILS|INVLN_DEDN_PAYEE_DETAILS|GMAC Collections|Employer Payee for Processing
 Fee|||||||||
MERGE|ComponentDetail|VISION|ID_GARN_IDSD_6485851|USA LDG|ID_GARN_6485851|2022/01/03||Vision Creditor Debt|
INVLN_DEDN_SUPPORT_DATA|INVLN_DEDN_SUPPORT_DATA|||2022/01/03||Tennessee garnishment||Y|Tennessee|||
MERGE|ComponentDetail|VISION|ID_GARN_IDD_6485851|USA LDG|ID_GARN_6485851|2022/01/03||Vision Creditor Debt|
HRX_US_INV_DEDN_DATA|HRX_US_INV_DEDN_DATA|||||||||4734333|1|DocID-12345
```

The parent Card Component is identified on the ComponentDetails records by the DirCardCompId(SourceSystemId) attribute, which has a value that matches the SourceSystemId on the CardComponent record.

Value Definitions

Use the CalculationValueDefinition and EnterableValueDefinition record types to provide override values for the value definitions.

In this example the following value definitions are overridden:

Value Definition	Value
Tennessee Garnishment Order Rate	.25
Tennessee Garnishment Total Owed Amount	7845.22
Tennessee Garnishment Number of Dependents	2

Note: Refer to the Cloud Customer Connect topic: *Utilize the US Involuntary Deductions Value Definitions* BI publisher report, to identify all value definitions for the Garnishment or Creditor debt element.

```
METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|ID_GARN_VD1_6485851|USA LDG|ID_GARN_6485851|2022/01/03||Vision
 Creditor Debt|Tennessee Garnishment Order Rate
MERGE|CalculationValueDefinition|VISION|ID_GARN_VD2_6485851|USA LDG|ID_GARN_6485851|2022/01/03||Vision
 Creditor Debt|Tennessee Garnishment Total Owed Amount
MERGE|CalculationValueDefinition|VISION|ID_GARN_VD3_6485851|USA LDG|ID_GARN_6485851|2022/01/03||Vision
 Creditor Debt|Tennessee Garnishment Number of Dependents

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|EffectiveEndDate|Value1
MERGE|EnterableCalculationValue|VISION|ID_GARN_VD1_ECV_6485851|USA LDG|ID_GARN_VD1_6485851|2022/01/03||0.25
MERGE|EnterableCalculationValue|VISION|ID_GARN_VD2_ECV_6485851|USA LDG|ID_GARN_VD2_6485851|2022/01/03||
7845.22
MERGE|EnterableCalculationValue|VISION|ID_GARN_VD3_ECV_6485851|USA LDG|ID_GARN_VD3_6485851|2022/01/03||2
```

The parent Card Component is identified on the CalculationValueDefinitions records by the SourceId(SourceSystemId) attribute which has a value that matches the SourceSystemId on the CardComponent record.

The parent for each Calculation Value Definition is identified on each EnterableCalculationValue record using the ValueDefnId(SourceSystemId) attribute, which has a value that matches the SourceSystemID attribute on the parent CalculationValueDefinition record.

Related Topics

- Overview of Involuntary Deductions for US Employees
- Load Involuntary Deduction Calculation Cards for US Employees
- Load a Card Component for US Involuntary Deductions

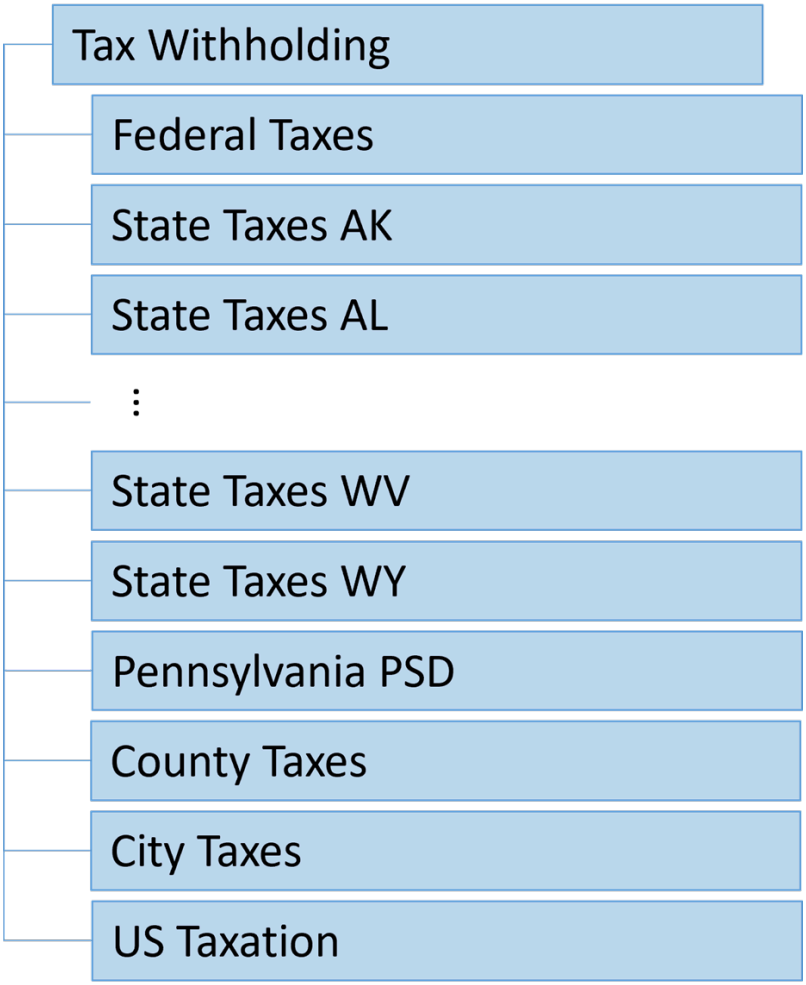
Tax Withholding Information - Dynamic Card

Overview of US Tax Withholding Information - Dynamic Card

Tax Withholding calculation cards store component information for employee federal, state, and local tax forms supported, and calculation overrides for the United States.

Calculation card components are not needed to calculate taxes. If a card component has not been created, or the component was created with no values or missing values, default values will be used for tax calculation.

The Tax Withholding HCM Data Loader object hides the complexity of the underlying calculation card structure. You'll find a single Tax Withholding business object for the different types of tax: federal, county, city etc. There are separate components for Federal, each State, and Local Tax in the Tax Withholding business object, ensuring each tax card component only includes attributes relevant to that component.



Tax Withholding

The Tax Withholding component is used to identify who the card is for. It should always be supplied when creating or maintaining existing cards.

Federal Taxes

The Federal Taxes card component is used to capture information that will impact Federal Tax calculations for the employee.

Each Tax Withholding calculation card should have one Federal Taxes card component.

State Taxes

The State Taxes components are used to capture information that will impact State Tax calculations for the employee.

A State Tax card component may be used for each US state that has a resident tax or work tax.

Pennsylvania PSD

The Pennsylvania PSD card component is used to capture information specific to Pennsylvania PSD. When a Pennsylvania residency certificate is received from an employee, create a single Pennsylvania PSD for the employee.

County Taxes

The County Taxes card component is used to capture information for the related resident or work County Tax calculations for the employee. A County Tax card component may be used for each US County that has a resident tax or work tax and requires an override. There is only one County Taxes component in the Tax Withholding object hierarchy, you specify the state and county for the attributes you are updating on the County Taxes component.

City Taxes

The City Taxes card component is used to capture information that impacts city tax calculations for the employee. A City Tax card component may be used for each US City that has a resident tax or work tax and requires and override.

There is only one City Taxes component in the Tax Withholding object hierarchy, you specify the state, county, and city for the attributes you are updating on the City Taxes component.

US Taxation

You should create a US Taxation card component for each tax reporting unit for the employee.

Considerations and Prerequisites

When you create your employees with the product license set to Payroll or Payroll Interface, Tax Withholding calculation cards are auto-generated for your US new hires with an association to their (Tax Reporting Unit) TRU, if the employee is only assigned to only one TRU. In other cases, the Tax Withholding card association may need to be created manually. A Federal component is also auto-generated for new hires. You then need to do one of the following:

- Allow employees to update their components using self-service
- Enter the updates manually based on the forms received from the employee
- Load the information using HDL based on the forms received from the employee.

For guidance on maintaining autogenerated cards refer to the Guidelines for Maintaining Tax Withholding Cards for the United States topic.

If your employees were created in the Oracle Human Capital Management Cloud without a Payroll or Payroll Interface license, then you need to create the complete Tax Withholding card. Refer to the Guidelines for Creating a Tax Withholding Card for the United States topic.

Guidelines for Maintaining Tax Withholding Cards for the United States

If your Tax Withholding cards need to be created from scratch refer instead to the Guidelines for Creating a Tax Withholding Card for the United States topic.

Tax Withholding

If you're updating tax information for an existing Tax Withholding calculation card always include the Tax Withholding record to group other related data supplied in the file. The effective start date you specify for the Tax Withholding record must be the original effective start date of the card.

When supplying the Tax Withholding record use the TaxWithholding file discriminator. For autogenerated cards you're unlikely to know the source key that identifies the record, so use these attributes to identify the card.

Attribute	Description
AssignmentNumber, or PayrollRelationshipNumber	Identify the employee the Tax Withholding card is for.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
CardSequence	The number that uniquely identifies the Tax Withholding card for the employee.
EffectiveStartDate	The start date of the card, typically the employee's hire date.

For example:

```
METADATA|TaxWithholding|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate|
SourceRef001=AssignmentNumber
MERGE|TaxWithholding|US LDG|E12833|1|2017/01/01|E12833
```

Federal Taxes

If correcting the existing Federal Taxes card component, the effective start date must be the original start date of the Federal Taxes. You can load date-effective history, or update Federal Taxes, for example when a new override value needs to be populated. When supplying the Federal Taxes record use the FederalTaxes file discriminator. For autogenerated cards you're unlikely to know the source key that identifies the record, so use these attributes to identify the card.

Attribute	Description
AssignmentNumber, or PayrollRelationshipNumber	Identify the employee the Tax Withholding card is for.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
CardSequence	The number that uniquely identifies the Tax Withholding card for the employee.
EffectiveStartDate	The start date of the card, typically the employee's hire date.

This example updates an existing Federal Taxes card component which starts on 1st Jan 2017 with the Tax Withholding card. The update to the Federal Taxes card component is made on 1st Jan 2023 to populate field values.

```
METADATA|TaxWithholding|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate
MERGE|TaxWithholding|US LDG|E12833|1|2017/01/01

METADATA|FederalTaxes|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate|
ExemptfromMedicare|RegularAmount
MERGE|FederalTaxes|US LDG|E12833|1|2023/03/01|Y|550
```

State Taxes

There is a separate component in the Tax Withholding object hierarchy for each US state. They include only the attributes specific to that state. When supplying State Taxes records use the file discriminator specific to the state you are providing data for, such as StateTaxesOH for Ohio. Use the View Business Objects task to identify the file discriminator and available attributes for each state.

> **Note:** State Withholding Overrides are in separate regions for Residential and Nonresidential. We use the same field names in each region. When loading these values, the residential fields will not have a prefix. The Nonresidential fields will have a prefix and will be loaded as: NonresidentialWithholdingOverrides_<field name>.

For autogenerated cards you're unlikely to know the source key that identifies the record, so use these attributes to identify the card.

Attribute	Description
AssignmentNumber, or PayrollRelationshipNumber	Identify the employee the Tax Withholding card is for.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
CardSequence	The number that uniquely identifies the Tax Withholding card for the employee.
EffectiveStartDate	The start date of the state taxes.

This example uses user keys to identify the Tax Withholding card and State Taxes. It creates State Taxes for California from 1st Jan 2023, setting the following fields:

Field	Value
Filing Status	Single
Number of Regular Withholding Allowances	1
Total Number of Allowances	1

```
METADATA|TaxWithholding|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate|
SourceRef001=AssignmentNumber
MERGE|TaxWithholding|US LDG|E12833|1|2017/01/01|E12833

METADATA|StateTaxesCA|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate|
FilingStatus|NumberofRegularWithholdingAllowances|TotalNumberofAllowances
MERGE|StateTaxesCA|US LDG|E12833|1|2023/01/01|Single|1|1
```

Pennsylvania PSD

When supplying Pennsylvania PSD use the PennsylvaniaPSD file discriminator. The following attributes are always required. Use the View Business Objects task to find other optional attributes for Pennsylvania PSD.

Attribute	Description
AssignmentNumber, or PayrollRelationshipNumber	Identify the employee the Tax Withholding card is for.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
CardSequence	The number that uniquely identifies the Tax Withholding card for the employee.
TaxReportingUnit	The tax reporting unit tied to the employee residency certificate.
EffectiveStartDate	The start date of the county taxes or update to the county taxes.

This example creates Pennsylvania PSD on the 1st Jan 2017 referring to the existing Tax Withholding card by user keys. The following field are set:

Field	Value
Tax Reporting Unit	US TRU
Resident PSD Code	10404
Resident School District	Gettysburg ASD
Work PSD Code	10404
Work School District	Gettysburg ASD

```
METADATA|TaxWithholding|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate|
SourceRef001=AssignmentNumber
MERGE|TaxWithholding|US LDG|E91747|1|2017/01/01|E91747

METADATA|PennsylvaniaPSD|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate|
TaxReportingUnit|ResidentPSDCode|ResidentSchoolDistrict|WorkPSDCode|WorkSchoolDistrict
MERGE|PennsylvaniaPSD|US LDG|E91747|1|2017/01/01|US TRU|10404|Gettysburg ASD|10404|Gettysburg ASD
```

County Taxes

When supplying County Taxes use the CountyTaxes file discriminator. The following attributes are always required. Use the View Business Objects task to find other optional attributes for County Taxes.

Attribute	Description
AssignmentNumber, or PayrollRelationshipNumber	Identify the employee the Tax Withholding card is for.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
CardSequence	The number that uniquely identifies the Tax Withholding card for the employee.
State	The two-letter abbreviation for the state, such as OH for Ohio.
County	The name of the county. Alternatively, supply the geocode for the county.
EffectiveStartDate	The start date of the county taxes or update to the county taxes.

This example identifies the existing Tax Withholding card and new County Taxes card component by user keys. It creates County Taxes from the 1st January 2023 setting the following fields:

Field	Value
State	KY
County	Grant
Resident County Tax Regular Amount	25

```
METADATA|TaxWithholding|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate|
SourceRef001=AssignmentNumber
MERGE|TaxWithholding|US LDG|E12833|1|2017/01/01|E12833

METADATA|CountyTaxes|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate|State|County|
ResidentCountyTaxRegularAmount
MERGE|CountyTaxes|US LDG|E12833|1|2023/01/01|KY|Grant|25
```

City Taxes

When supplying City Taxes use the CityTaxes file discriminator. The following attributes are always required. Use the View Business Objects task to find other optional attributes for City Taxes.

Attribute	Description
AssignmentNumber, or PayrollRelationshipNumber	Identify the employee the Tax Withholding card is for.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
CardSequence	The number that uniquely identifies the Tax Withholding card for the employee.
State	The two-letter abbreviation for the state, such as OH for Ohio.
County	The name of the county. Alternatively, supply the geocode for the county.
City	The name of the city. Alternatively, supply he geocode for the city.
EffectiveStartDate	The start date of the county taxes or update to the county taxes.

This example identifies the existing Tax Withholding card and new City Taxes card component by user keys. It creates City Taxes on the 1st Jan 2023 with the following values:

Field	Value
State	PA
County	Philadelphia
City	Philadelphia
Nonresident City Tax Regular Amount	200
Nonresident City Tax Supplemental Amount	100

```
METADATA|TaxWithholding|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate|
SourceRef001=AssignmentNumber
MERGE|TaxWithholding|US LDG|E12833|1|2017/01/01|E12833

METADATA|CityTaxes|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate|State|County|
City|NonresidentCityTaxRegularAmount|NonresidentCityTaxSupplementalAmount
MERGE|CityTaxes|US LDG|E12833|1|2023/01/01|PA|Philadelphia|Philadelphia|200|100
```

US Taxation

When supplying US Taxation records use the USTaxation file discriminator. The following attributes are required or commonly used. Use the View Business Objects task to find other optional attributes for US Taxation.

Create or Update

Attribute	Description
AssignmentNumber, or PayrollRelationshipNumber	Identify the employee the Tax Withholding card is for.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
CardSequence	The number that uniquely identifies the Tax Withholding card for the employee.
EffectiveStartDate	The start date of the card, typically the employee's hire date.
AssociationAssignmentNumber	The Tax Withholding card is identified by the employee's primary assignment. If additional assignments are to be associated with the card supply the assignment number to this attribute.
TaxReportingUnit	The tax reporting unit associated to the US Taxation Component.

This example corrects the autogenerated US Taxation:

```
METADATA|TaxWithholding|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate
MERGE|TaxWithholding|US LDG|E12833|1|2017/01/01

METADATA|USTaxation|LegislativeDataGroupName|AssignmentNumber|CardSequence|EffectiveStartDate|
TaxReportingUnit|StatutoryEmployee|PrimaryWorkAddress|StateforDisabilityCalculation|
StateforUnemploymentCalculation|CumulativeTaxation
MERGE|USTaxation|US LDG|E12833|1|2017/01/01|US TRU|Y|DEFAULT|KY|KY|N
```

End Date Association:

Attribute	Description
AssignmentNumber, or PayrollRelationshipNumber	Identify the employee the Tax Withholding card is for.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
CardSequence	The number that uniquely identifies the Tax Withholding card for the employee.
EffectiveStartDate	The start date of the card, typically the employee's hire date.
EffectiveEndDate	The end date of the record.
AssociationTaxReportingUnitName	Tax Reporting Unit associated to the US Taxation Component
ReplaceLastEffectiveEndDate	Y

Attribute	Description
TaxReportingUnit	The tax reporting unit associated to the US Taxation Component.

End Date Association Details:

Attribute	Description
AssignmentNumber, or PayrollRelationshipNumber	Identify the employee the Tax Withholding card is for.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
CardSequence	The number that uniquely identifies the Tax Withholding card for the employee.
EffectiveStartDate	The start date of the card, typically the employee's hire date.
EffectiveEndDate	The end date of the association assignment.
AssociationAsignmentNumber	Assignment number being end dated.
ReplaceLastEffectiveEndDate	Y
TaxReportingUnit	The tax reporting unit associated to the US Taxation Component.

Delete Association:

Attribute	Description
AssignmentNumber, or PayrollRelationshipNumber	Identify the employee the Tax Withholding card is for.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
CardSequence	The number that uniquely identifies the Tax Withholding card for the employee.
EffectiveStartDate	The start date of the card, typically the employee's hire date.
AssociationTaxReportingUnitName	The tax reporting unit on the US Taxation Component being deleted.
TaxReportingUnit	The tax reporting unit associated to the US Taxation Component.

Delete Association Details:

Attribute	Description
AssignmentNumber, or PayrollRelationshipNumber	Identify the employee the Tax Withholding card is for.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
CardSequence	The number that uniquely identifies the Tax Withholding card for the employee.
EffectiveStartDate	The start date of the card, typically the employee's hire date.

Attribute	Description
AssociationAsignmentNumber	Assignment number being deleted.
TaxReportingUnit	The tax reporting unit associated to the US Taxation Component.

Guidelines for Creating a Complete Tax Withholding Card for the United States

If your employees were created in the Oracle Human Capital Management Cloud without a Payroll or Payroll Interface license, then you need to create the complete Tax Withholding card.

When creating a complete card, it's recommended that you supply source keys to uniquely identify each record. You'll need to supply the following components of the Tax Withholding object hierarchy:

- Tax Withholding
  - Federal Taxes
  - State Taxes for each US state where the employee has submitted a tax withholding form.
  - Pennsylvania PSD if the employee lives or works in Pennsylvania and submitted a residency certificate
  - County and City Taxes if any field updates are required.
  - US Taxation

Tax Withholding

When supplying the Tax Withholding record use the TaxWithholding file discriminator, the following attributes are always required:

Attribute	Description
SourceSystemId, SourceSystemOwner	Supply a source key to uniquely identify the record.
LegislativeDataGroupName	The name of the legislative data group for the calculation card. This field appears on all records within your TaxWitholding.dat file and should have the same value for a single Tax Withholding card.
CardSequence	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee.
AssignmentId(SourceSystemId), AssignmentId(SourceSystemOwner)	A reference to the employee's assignment. These attributes are used to supply the source key of the employee assignment. Alternatively, supply AssignmentNumber or PayrollRelationshipNumber
EffectiveStartDate	The start date of the card, typically the employee's hire date.

Federal Taxes

When supplying the Federal Taxes record use the FederalTaxes file discriminator the following attributes are always required. Use the **View Business Objects** task to find other optional attributes, such as overrides for Federal Taxes.

Attribute	Description
SourceSystemId, SourceSystemOwner	Supply a source key to uniquely identify the record.
LegislativeDataGroupName	The name of the legislative data group for the calculation card. This field appears on all records within your TaxWitholding.dat file and should have the same value for a single Tax Withholding card.
DirCardId(SourceSystemId)	This identifies the parent record. Specify the SourceSystemId value from the parent TaxWithholding record.
EffectiveStartDate	The start date of the card, typically the employee's hire date, or the update date of the Federal Taxes record.

State Taxes

There is a separate component in the Tax Withholding object hierarchy for each US state. They include only the attributes specific to that state. When supplying State Taxes records, use the file discriminator specific to the state you are providing data for, for example: **StateTaxesOH** for Ohio. Use the **View Business Objects** task to identify the file discriminator and available attributes for each state.

The following attributes are required for all state tax records.

Attribute	Description
SourceSystemId, SourceSystemOwner	Supply a source key to uniquely identify the record.
LegislativeDataGroupName	The name of the legislative data group for the calculation card. This field appears on all records within your TaxWitholding.dat file and should have the same value for a single Tax Withholding card.
DirCardId(SourceSystemId)	This identifies the parent record. Specify the SourceSystemId value from the parent TaxWithholding record.
EffectiveStartDate	The start date of the card, typically the employee's hire date, or the update date of the Federal Taxes record.

Pennsylvania PSD

When supplying Pennsylvania PSD use the PennsylvaniaPSD file discriminator the following attributes are always required. Use the View Business Objects task to find other optional attributes for Pennsylvania PSD.

Attribute	Description
SourceSystemId, SourceSystemOwner	Supply a source key to uniquely identify the record.
LegislativeDataGroupName	The name of the legislative data group for the calculation card.
DirCardId(SourceSystemId)	This identifies the parent record. Specify the SourceSystemId value from the parent TaxWithholding record.
AssignmentId(SourceSystemId), AssignmentId(SourceSystemOwner)	A reference to the employee's assignment. These attributes are used to supply the source key of the employee assignment. Alternatively, supply AssignmentNumber or PayrollRelationshipNumber. Without this reference the Tax Reporting Unit validation will fail.

Attribute	Description
TaxReportingUnit	The tax reporting unit associated to the US Taxation Component
EffectiveStartDate	The start date of the Pennsylvania PSD record.

County Taxes

The County Taxes card component is used to capture information that impact County Tax calculations for the employee. Not all states require a County Taxes card component. When supplying County Taxes use the CountyTaxes file discriminator the following attributes are always required. Use the **View Business Objects** task to find other optional attributes for County Taxes.

Attribute	Description
SourceSystemId, SourceSystemOwner	Supply a source key to uniquely identify the record.
LegislativeDataGroupName	The name of the legislative data group for the calculation card. This field appears on all records within your TaxWitholding.dat file and should have the same value for a single Tax Withholding card.
DirCardId(SourceSystemId)	This identifies the parent record. Specify the SourceSystemId value from the parent TaxWithholding record.
State	The two-letter abbreviation for the state, such as OH for Ohio.
County	The name of the county. Alternatively, supply the geocode for the county.
EffectiveStartDate	The start date of the card, typically the employee's hire date, or the update date of the Federal Taxes record.

City Taxes

The City Taxes card component is used to capture information that impact city tax calculations for the employee. Not all states require a City Taxes card component. When supplying City Taxes use the CityTaxes file discriminator the following attributes are always required. Use the **View Business Objects** task to find other optional attributes for City Taxes.

Attribute	Description
SourceSystemId, SourceSystemOwner	Supply a source key to uniquely identify the record.
LegislativeDataGroupName	The name of the legislative data group for the calculation card. This field appears on all records within your TaxWitholding.dat file and should have the same value for a single Tax Withholding card.
DirCardId(SourceSystemId)	This identifies the parent record. Specify the SourceSystemId value from the parent TaxWithholding record.
State	The two-letter abbreviation for the state, such as OH for Ohio.
County	The name of the county. Alternatively, supply the geocode for the county.
City	The name of the city. Alternatively, supply the geocode for the city.

Attribute	Description
EffectiveStartDate	The start date of the card, typically the employee's hire date, or the update date of the Federal Taxes record.

US Taxation

Supply a US Taxation card component and override values for each tax reporting unit for the employee. When supplying US Taxation records use the USTaxation file discriminator the following attributes are required or commonly used. Use the **View Business Objects** task to find other optional attributes for US Taxation.

Attribute	Description
SourceSystemId, SourceSystemOwner	Supply a source key to uniquely identify the record.
LegislativeDataGroupName	The name of the legislative data group for the calculation card. This field appears on all records within your TaxWitholding.dat file and should have the same value for a single Tax Withholding card.
DirCardId(SourceSystemId)	This identifies the parent record. Specify the SourceSystemId value from the parent TaxWithholding record.
AssignmentId(SourceSystemId), AssignmentId(SourceSystemOwner)	A reference to the employee's assignment to associate with the card. These attributes are used to supply the source key of the employee assignment. Alternatively, supply AssignmentNumber or PayrollRelationshipNumber.
TaxReportingUnit	The tax reporting unit associated to the US Taxation Component
EffectiveStartDate	The start date of the card, typically the employee's hire date, or the update date of the Federal Taxes record.

Examples of Creating a Complete Tax Withholding Card for the United States

This example creates a single Tax Withholding card for employee assignment E91413 starting on 1st Jan 2018 using source keys.

There are two Federal Taxes records, the first starting with the Tax Withholding card, the second is an update on the 1st Jan 2023 to capture new override values.

There are state, county and city tax records for Ohio and Kentucky.

```
METADATA|TaxWithholding|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|EffectiveStartDate|
CardSequence|AssignmentId(SourceSystemId)|AssignmentId(SourceSystemOwner)|SourceRef001=AssignmentNumber
MERGE|TaxWithholding|VISION|TW_E91413|US LDG|2018/01/01|1|ASG-91413|EMP|E91413

METADATA|FederalTaxes|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|FilingStatus|MultipleJobs|ExtraWithholding|ExemptfromFederalIncomeTax|
NonresidentAlien|QualifyingDependentsAmount|OtherDependentsAmount|TotalDependentsAmount|OtherIncomeAmount|
DeductionsAmount|ExemptfromMedicare|ExemptfromFederalUnemployment|ExemptfromSocialSecurity|
ExemptfromWageAccumulation|EnforceFederalIncomeTaxLookbackRule|RegularAmount|RegularRate|SupplementalAmount|
SupplementalRate
```

```
MERGE|FederalTaxes|VISION|TW_E91413_FT|US LDG|TW_E91413|2018/01/01|2022/12/31|Head of household||
100|||||||||||||||
MERGE|FederalTaxes|VISION|TW_E91413_FT|US LDG|TW_E91413|2023/01/01|||||||10|20|30|40|50|N|N|N|N|N||||

METADATA|StateTaxesCA|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|FilingStatus|NumberofRegularWithholdingAllowances|TotalNumberofAllowances
MERGE|StateTaxesCA|VISION|TW_E91413_STCA|US LDG|TW_E91413|2018/01/01|Single|1|1

METADATA|StateTaxesKY|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|AdditionalTaxAmount
MERGE|StateTaxesKY|VISION|TW_E91413_STKY|US LDG|TW_E91413|2018/01/01|10

METADATA|CountyTaxes|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|State|County|ExemptfromCountyIncomeTax
MERGE|CountyTaxes|VISION|TW_E91413_CT117|US LDG|TW_E91413|2018/01/01|KY|Boyd|Y

METADATA|CityTaxes|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|State|County|City|AdditionalTaxAmount
MERGE|CityTaxes|VISION|TW_E91413_CI1725|US LDG|TW_E91413|2018/01/01|KY|Madison|Berea|10

METADATA|USTaxation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|AssignmentId(SourceSystemId)|AssignmentId(SourceSystemOwner)|TaxReportingUnit|
AssociationAssignmentNumber|StatutoryEmployee|PrimaryWorkAddress|StateforDisabilityCalculation|
StateforUnemploymentCalculation|CumulativeTaxation
MERGE|USTaxation|VISION|TW_E91413_UST|US LDG|TW_E91413|2018/01/01|ASG-91413|EMP|US TRU|E91413|Y|DEFAULT|KY|
KY|Y
```

This example creates a complete Tax Withholding component for Pennsylvania PSD as an addition to the example above.

```
METADATA|PennsylvaniaPSD|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|AssignmentId(SourceSystemId)|AssignmentId(SourceSystemOwner)|
TaxReportingUnit|ResidentPSDCode|ResidentSchoolDistrict|WorkPSDCode|WorkSchoolDistrict
MERGE|PennsylvaniaPSD|VISION|TW_E91413_PSD|US LDG|TW_E91413|2017/01/01|ASG-81416|EMP|US TRU|10404|Gettysburg
 ASD|10404|Gettysburg ASD
```

Tax Withholding for Pension and Annuity Payments

Overview of US Tax Withholding for Pension and Annuity Payments

Tax Withholding for Pension and Annuity Payments cards store all the information required to accurately compute federal and state tax for a retiree.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, Tax Withholding calculation cards are auto-generated for your US new hires. You then need to:

- Update Retiree US Taxation and Federal Tax
- Create State Tax applicable to the state
- Create a tax reporting unit association for the Retiree US Taxation card component

If your retirees were created in the Oracle Human Capital Management Cloud without a Payroll or Payroll Interface license then you need to create the complete Tax Withholding for Pension and Annuity Payments card.

Tax Withholding for Pension and Annuity Payments Record Types

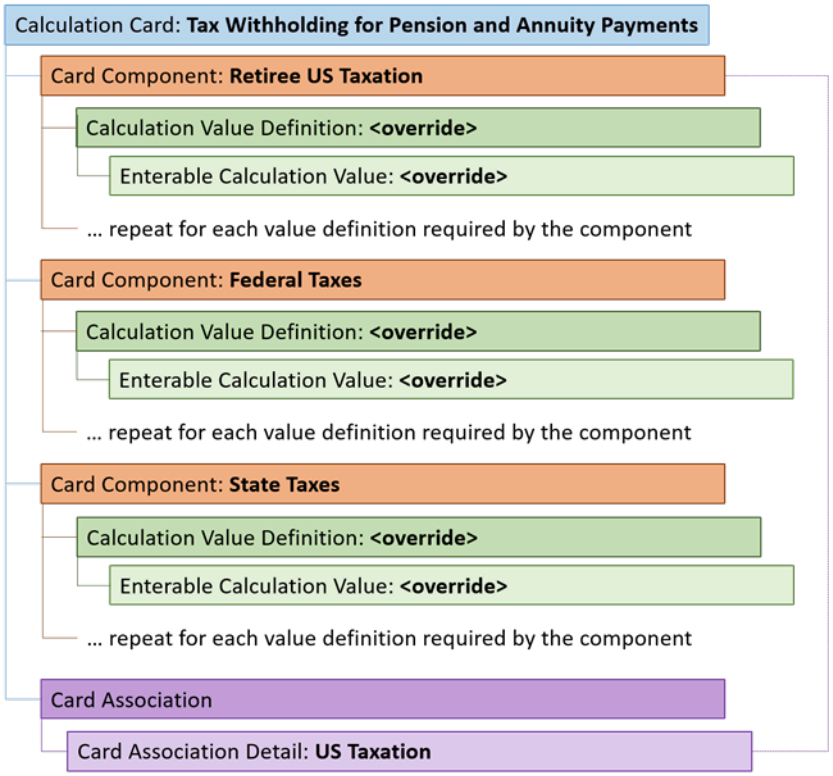
The Tax Withholding for Pension and Annuity Payments card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various localization requirements.

The Tax Withholding for Pension and Annuity Payments card utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Calculation Value Definition	Allows the creation of value definitions so that overriding values can be specified on the card component. Details of the specific value definitions are provided in the following sections.	CalculationValueDefinition
Enterable Calculation Value	Used to specify an overriding value for each calculation value definition.	EnterableCalculationValue
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Tax Withholding for Pension and Annuity Payments Card Hierarchy

The hierarchy of Calculation Card components applicable to Tax Withholding for Pension and Annuity Payments are described here:



Related Topics

- Guidelines for Loading US Tax Withholding for Pension and Annuity Payments
- Guidelines for Loading Retiree US Taxation for Pension and Annuity Payments
- Guidelines for Loading US Federal Tax for Pension and Annuity Payments
- Guidelines for Loading US State Tax for Pension and Annuity Payments
- Associate Retiree US Taxation for Pension and Annuity Payments

Guidelines for Loading US Tax Withholding for Pension and Annuity Payments

Provide one Calculation Card record for every US retiree you are maintaining Tax Withholding for Pension and Annuity Payments data for.

Even if you are updating an existing calculation card and the calculation card itself is not being updated, still include the calculation card record to group other related data supplied in the file. The effective start date you specify for the Tax Withholding for Pension and Annuity Payments card must be the original effective start date of the card.

Tax Withholding for Pension and Annuity Payments Calculation Card Attributes

The Tax Withholding for Pension and Annuity Payments calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Tax Withholdingfor Pension and Annuity Payments calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Tax Withholding for Pension and Annuity Payments'
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the retiree this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's retirement date.

Examples of supplying these attributes are supplied in the card component topics.

Related Topics

- Overview of US Tax Withholding for Pension and Annuity Payments

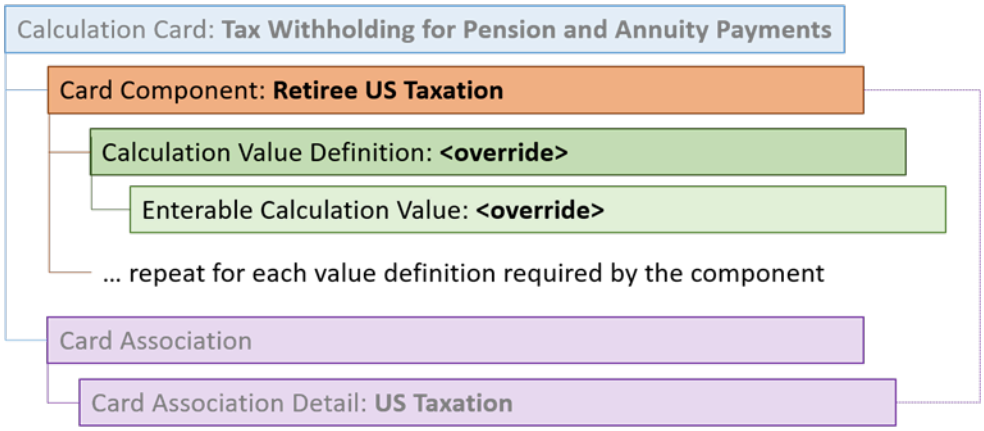
Guidelines for Loading Retiree US Taxation for Pension and Annuity Payments

Supply a Retiree US Taxation card component and related override values for each tax reporting unit for the employee.

If you are updating existing components, the effective start date you supply must be the original start date of the Retiree US Taxation card component.

> **Note:** The effective start date of the Retiree US Taxation component should match the start date of the Tax Withholding for Pension and Annuity Payments card.

Retiree US Taxation Card Component Hierarchy



The Retiree US Taxation card component uses value definitions to capture override values. Value definitions are supplied using the Calculation Value Definition and Enterable Calculation Value record types. The following sections describe how to supply valid file lines for these record types.

Card Component Attributes for Retiree US Taxation

The Retiree US Taxation Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Retiree US Taxation card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardDefinitionName	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent **Tax Withholding for Pension and Annuity Payments** calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the RetireeUS Taxation card component, typically the employee's retirement date. This must be the same as the EffectiveStartDate on the Tax Withholdingfor Pension and Annuity Payments calculation card. If updating an existing RetireeUS Taxation card component, the effective start date must be original start date of the component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName	N/A	The component definition name. Specify 'Retiree US Taxation'.
ComponentSequence	N/A	The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.
Context1	N/A	The surrogate ID of the employee's tax reporting unit. It is an application generated value to uniquely identify the tax reporting unit. **Note:** This attribute is required.

These attributes are supplied against the CardComponent file discriminator and must be supplied along with a CalculationCard record for the parent Tax Withholding card.

Retiree US Taxation Value Definitions

The Retiree US Taxation card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Cumulative Taxation	To indicate if the person's taxation should be on a cumulative basis. Supply Y or N.
Statutory Employee	Identifies the retiree as statutory. Supply Y or N.
Primary Work Address	The address of the work location the retiree is assigned to. Typically their home address.

Calculation Value Definition Attributes for Retiree US Taxation

The Calculation Value Definition record type specifies the name of the value definition you are supplying an override value for.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Retiree US Taxation card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent RetireeUS Taxation card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent RetireeUS Taxation card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent **Retiree US Taxation** card component and a CalculationCard record for the owning Tax Withholding for Pension and Annuity Payments card.

Enterable Calculation Value Attributes for US Taxation

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveEndDate	AssignmentNumber	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes.
Value1	N/A	The value for the value definition identified by the parent calculation value definition record. Unlike other calculation cards, if you supply a value definition but have no value for it supply '-999999999' to indicate a null value.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Related Topics

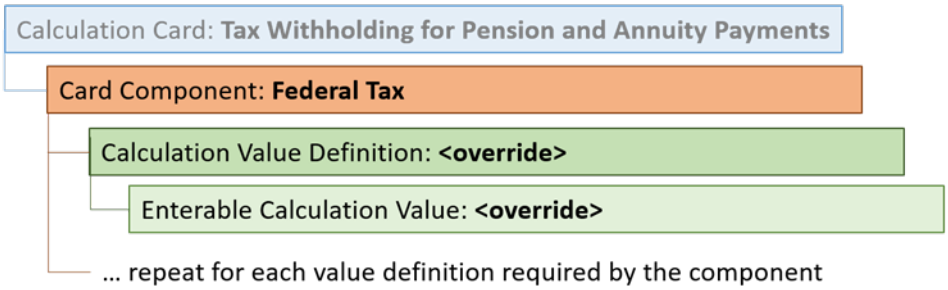
- Overview of US Tax Withholding for Pension and Annuity Payments

Guidelines for Loading US Federal Tax for Pension and Annuity Payments

The Federal Tax card component is used to capture information that will impact Federal Tax calculations for the retiree.

Each Tax Withholding for Pension and Annuity Payments calculation card should have one Federal Tax card component.

Federal Tax Card Component Hierarchy for Tax Withholding for Pension and Annuity Payments



The Federal Tax card component uses value definitions to capture override values. Value definitions are supplied using the Calculation Value Definition and Enterable Calculation Value record types.

The following sections describe how to supply valid file lines for these record types.

Card Component Attributes for Federal Tax

The Federal Tax Card Component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Federal Tax card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	TThe name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Tax Withholding for Pension and Annuity Payments calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the Federal Tax card component, typically the employee's retirement date. This must be the same as the EffectiveStartDate on the Tax Withholding for Pension and Annuity Payments calculation card. If updating an existing Federal Tax card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName		The component definition name. Specify 'Federal Tax'.
ComponentSequence		The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.

These attributes are supplied against the CardComponent file discriminator and must be supplied along with a CalculationCard record for the parent Tax Withholding card.

Federal Tax Value Definitions

The Federal Tax card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Filing Status	Determines which tax chart to use when calculating the Federal Tax to be deducted from the retiree's pay.
Allowances	An exemption that reduces how much Federal Income Tax is deducted from the retiree's pension and annuity payments.
Additional Tax Amount	Additional tax amount to withhold for the employee's Federal Tax.
Exempt from Federal Income Tax	To indicate no Federal Income Tax withholding. Specify Y or N.
Exempt from Wage Accumulation	To indicate no Federal Income Tax withholding and no Federal Income Tax wage accumulation. Specify Y or N.
Regular Amount	Used to indicate a flat withholding amount for Regular pay runs.
Regular Rate	Used to indicate a percentage withholding for Regular pay runs.
Supplemental Amount	Used to indicate a flat withholding amount for Supplemental pay runs.
Supplemental Rate	Used to indicate a percentage withholding for Supplemental pay runs.
IRS Lock in Date	Date when IRS lock-in letter is effective.

Calculation Value Definition Attributes for Federal Tax for Retirees

The Calculation Value Definition record type specifies the name of the value definition you are supplying an override value for.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition..
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Federal Tax card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent Federal Tax card component or the date the calculation value definition starts, if later.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
DirCardCompDefName	N/A	The definition name of the parent Federal Tax card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent Federal Tax card component and a CalculationCard record for the owning Tax Withholding for Pension and Annuity Payments card.

Enterable Calculation Value Attributes for Federal Tax

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes.
Value1	N/A	The value for the value definition identified by the parent calculation value definition record. Unlike other calculation cards, if you supply a value definition but have no value for it supply '-999999999' to indicate a null value.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Related Topics

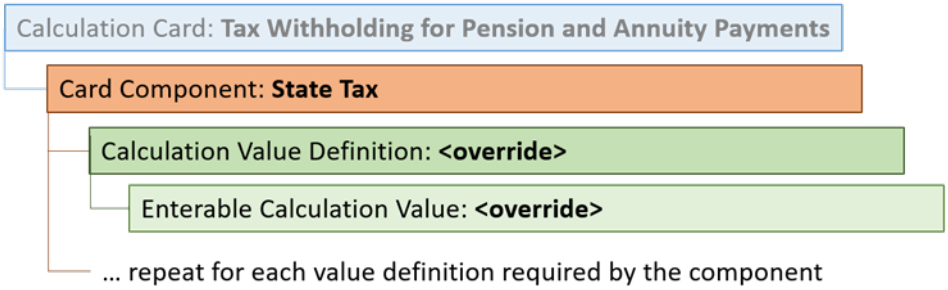
- Overview of US Tax Withholding for Pension and Annuity Payments

Guidelines for Loading US State Tax for Pension and Annuity Payments

The State Tax card component is used to capture information that impact state tax calculations for the retiree.

Each Tax Withholding for Pension and Annuity Payments calculation card should have one State Tax card component for each US state the retiree has a tax relationship with.

State Tax Card Component Hierarchy



The State Tax card component uses value definitions to capture override values. There are core value definitions which are applicable to all US states and additional value definitions which are state specific.

Value definitions are supplied using the Calculation Value Definition and Enterable Calculation Value record types. The following sections describe how to supply valid file lines for these record types.

Card Component Attributes for State Tax

The State Tax card component uses these attributes:

The State Tax card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the State Tax card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Tax Withholding for Pension and Annuity Payments calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the State Tax card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Tax Withholding for Pension and Annuity Payments calculation card. If updating an existing State Tax card component, the effective start date must be original start date of the component.
EffectiveEndDate	N/A	The end date is optional for the card component.
DirCardCompDefName		The component definition name. Specify 'State Tax'.
ComponentSequence		The number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exist. Not required when source keys are used.
Context1		The geocode number that identifies the US state. Refer to the Cloud Customer Connect topic Report: US State, County and City geographical codes.

These attributes are supplied against the CardComponent file discriminator and must be supplied along with a CalculationCard record for the parent Tax Withholding for Pension and Annuity Payments card.

State Tax Value Definitions

The State Tax card component uses value definitions to supply override values.

This table lists the value definitions that are common for all US states.

Global Value Definitions for State Tax

Value Definition Name	Functional Description
Filing Status	Determines which tax chart to use when calculating the Federal Tax to be deducted from the employees' pay.

Value Definition Name	Functional Description
Allowances	An exemption that reduces how much Federal Income Tax is deducted from the employee's pay. This field is for the 2019 and prior tax form.
Nonresident	Indicates if the employee is a nonresident of that state. Specify Y or N.
Voluntary Income Tax Withholding	For voluntary tax withholding for that state. Specify Y or N.
Resident Wage Accumulation	For use in determining withholding rules. Provide a lookup code from the lookup type HRX_US_STATE_WAGE_ACCUM.
Exempt from State Income Tax	Indicates an employee is exempt from this tax. Selecting this option does not withhold tax, but accumulates wages for the component. Specify Y or N.
Exempt from Wage Accumulation	Indicates an employee is exempt from this tax. Selecting this option will not withhold tax, and not accumulate wages for the component. Specify Y or N.
Exemption Amount	Claims an amount exempt from tax for that state.
Exemption for Military Spouse	A spouse of a service member retains residency in his or her home state for tax and voting purposes if he or she moves to another state to be with the service member who is in that state due to military orders. Thus, income earned in the work state by the spouse who is not a resident and is domiciled in another state is not subject to taxation by the work state. Specify Y or N.
Additional Tax Amount	Additional tax amount to withhold for the employee's state tax
Nonresident State Income Tax Supplemental Rate	The supplemental tax rate for the nonresident state tax.
Nonresident State Income Tax Regular Rate	The regular tax rate for the nonresident state tax.
Nonresident State Income Tax Regular Amount	The regular tax amount for the nonresident state tax.
Resident State Income Tax Supplemental Rate	The supplemental tax rate for the resident state tax.
Resident State Income Tax Regular Rate	The regular tax rate for the resident state tax.
Resident State Income Tax Regular Amount	The regular tax amount for the resident state tax.

Calculation Value Definition Attributes for State Tax

The Calculation Value Definition record type specifies the value definition name for the override value.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent State Tax card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent State Tax card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent State Tax card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.
Context1	N/A	The geocode number that identifies the US state. Specify the same value as provided to the Context1 attribute on the parent State Tax card component.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent **State Tax** card component and a CalculationCard record for the owning **Tax Withholding for Pension and Annuity Payments** card.

Enterable Calculation Value Attributes for State Tax

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnIdSourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes.
Value1		The value for the value definition identified by the parent calculation value definition record. Unlike other calculation cards, if you supply a value definition but have no value for it supply '-999999999' to indicate a null value.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Associate Retiree US Taxation for Pension and Annuity Payments

The card association record associates the Retiree Tax Withholding calculation card with a tax reporting unit.

Tax Withholding Card Association Hierarchy

The associated tax reporting unit is defined in the card association. The card association details records allows the Retiree US Taxation card component to be associated with the retiree's payroll assignment.

The following sections describe how to provide valid file lines for these record types.

Card Association Attributes for Tax Withholding

The Card Association record type associates a Tax Withholding card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Tax Withholdingfor Pension and Annuity Payments card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Tax Withholding for Pension and Annuity Payments calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	AssignmentNumber	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Tax Withholding for Pension and Annuity Payments.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Tax Withholding for Pension and Annuity Payments calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Retiree US Taxation

The card association details record associates the US Taxation card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	AssignmentNumber	The effective start date of the card association.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Identify the Retiree US Taxation card component this association is for.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

25 Loading Payroll Localization Data for France

Statutory Deductions

Overview of Statutory Deductions for France

Statutory Deduction cards store all the information required to accurately compute social insurance contributions, pension contributions and taxes.

One card must be created for each employee.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, employee Statutory Deduction cards are automatically created when a new employee is entered using the **New Hire** task with a set of default values specified at the Payroll Statutory Unit (PSU) or the Tax Reporting Unit (TRU) level. TRU association and association details for the employee are created automatically when the TRU is selected during the hire process.

However, there may be cases where this information must be loaded in bulk:

During data migration:

Employee social insurance information must be uploaded into Oracle Fusion Payroll, to ensure that contributions are calculated correctly. If HCM Data Loader is used to migrate employee records, a default statutory deduction card is automatically created. The default may not reflect the employee's actual social insurance information, and therefore the card must be updated.

Ongoing bulk updates:

- Bulk loading of new hire data: If you have to do a mass upload of New Hire information, a default Statutory Deductions card may be automatically generated (if the new hire records are created through HCM Data Loader). In this case, you need to update the default card with the correct information.

You must create element eligibility for element French Payroll Processing. This needs to be done once for each LDG that is used for entering person calculation cards. This task is required even if payroll is not processed with Oracle Fusion Payroll for France.

It is recommended to have a good understanding of the Statutory Deductions card and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS (France): HR Implementation and Functional Considerations Whitepaper (Doc ID 1542406.1).

Statutory Deduction Card Record Types

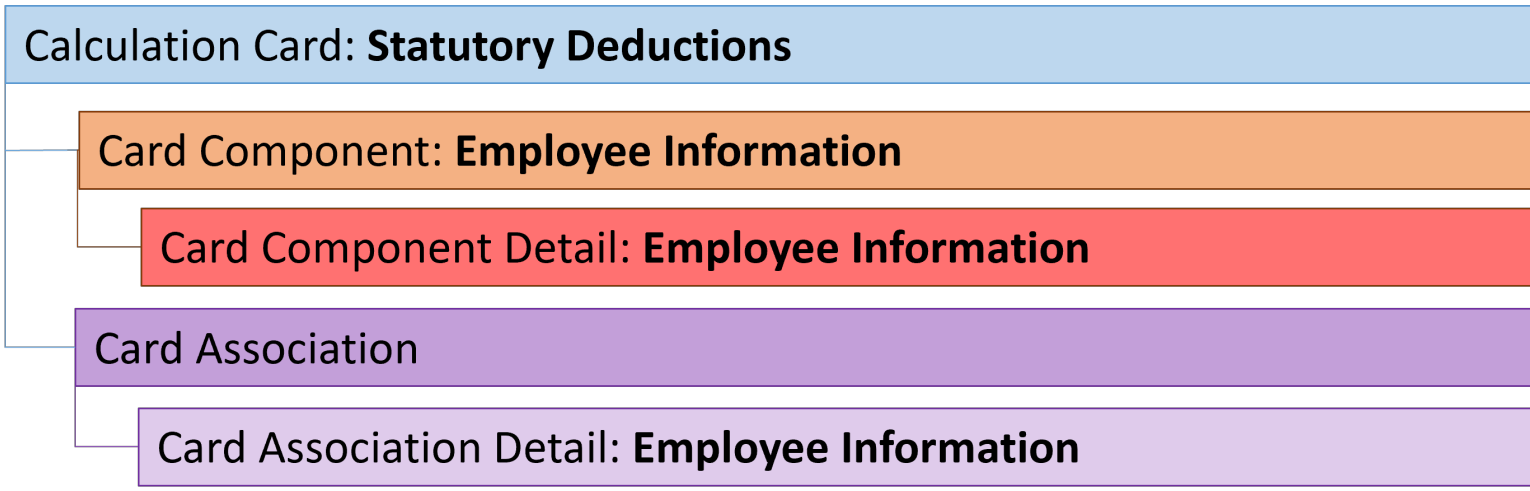
The Statutory Deductions card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The French Statutory Deductions utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card comonoent.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Statutory Deduction Calculation Card Hierarchy

The hierarchy of calculation card components applicable Statutory Deductions are described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Statutory Deductions for France

Guidelines for Loading Statutory Deductions for France

You must have one Calculation Card record for every French employee you are maintaining statutory deduction data for.

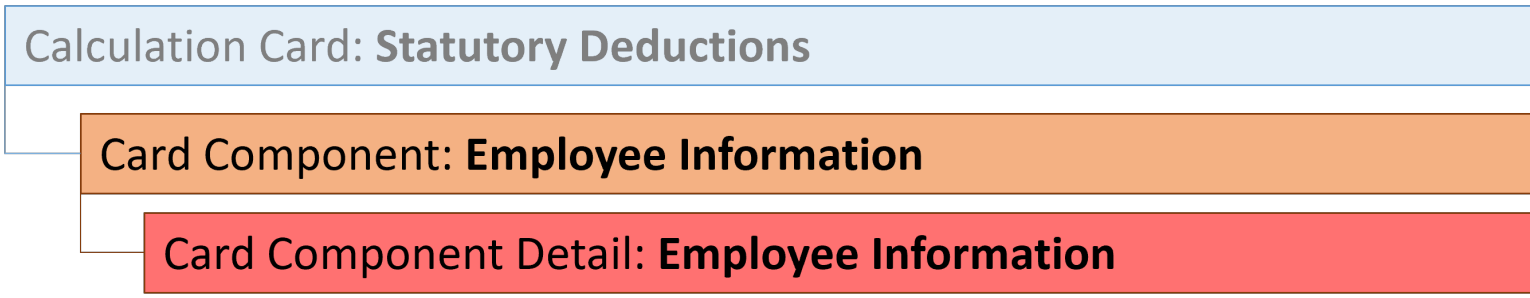
Even if you are updating an existing Statutory Deduction card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Statutory Deductions Calculation Card Attributes

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Statutory Deductions calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Statutory Deductions'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Employee Information Card Component

The Employee Information card component records all relevant social insurance information. It is created for you if the Statutory Deductions card is autogenerated. If you need to update the defaulted employee information, provide an Employee Information card component and related Component Detail for each employee.



The Employee Information card component uses the Employee Information flexfield context to capture data. Flexfield segments are uploaded using the Component Detail record type.

Card Component Attributes for Employee Information

The Employee Information card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee Information card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Employee Information card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Statutory Deductions calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Employee Information'.
ComponentSequence	N/A	Specify '1'.

Card Component Detail Attributes for Employee Information

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Employee Information (HRX_FR_EMPLOYEE_INFO)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory,	A unique identifier for the component detail. For new component detail records supply the

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Employee Information card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Information card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_FR_EMPLOYEE _INFO'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Statutory Deductions for France
- Guidelines for Loading Flexfield Data

Example of Updating Employee Information for French Statutory Deductions

This example updates the auto-generated Employee Information card component of the Statutory Deductions card for employee assignment E2323 on the 1st January 2023.

User keys are provided on the assumption that the Statutory Deductions card was autogenerated and therefore the source key values are unknown.

These values are updated:

Field	Value
Employee Scheme	PROFESSIONALIZATION_OVER_45
Local Scheme	OTHER_AREAS
Employee Pension Scheme	NON_MANAGER_APPRENTICE
Resident Abroad	Y
State Pension Contribution Base Full-Time	N
Complementary Pension Contribution Base Full-Time	Y

Use the CalculationCard.dat file to upload the Employee Information card component information with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|FR LDG|Statutory Deductions|2023/01/01|E2323|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|FR LDG|Statutory Deductions|2023/01/01|E2323|1|Employee Information|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer
 DF|_EMPLOYEE_SCHEME(Deduction Developer DF=HRX_FR_EMPLOYEE_INFO)|_LOCAL_SCHEME(Deduction Developer
 DF=HRX_FR_EMPLOYEE_INFO)|_EMPLOYEE_PENSION_SCHEME(Deduction Developer DF=HRX_FR_EMPLOYEE_INFO)|
_RESIDENT_ABROAD(Deduction Developer DF=HRX_FR_EMPLOYEE_INFO)|_STATE_PENSION(Deduction Developer
 DF=HRX_FR_EMPLOYEE_INFO)|_COMPLEMENTARY_PENSION(Deduction Developer DF=HRX_FR_EMPLOYEE_INFO)
MERGE|ComponentDetail|FR LDG|Statutory Deductions|2023/01/01|E2323|1|Employee Information|1|
HRX_FR_EMPLOYEE_INFO|HRX_FR_EMPLOYEE_INFO|PROFESSIONALIZATION_OVER_45|OTHER_AREAS|NON_MANAGER_APPRENTICE|Y|
N|Y
```

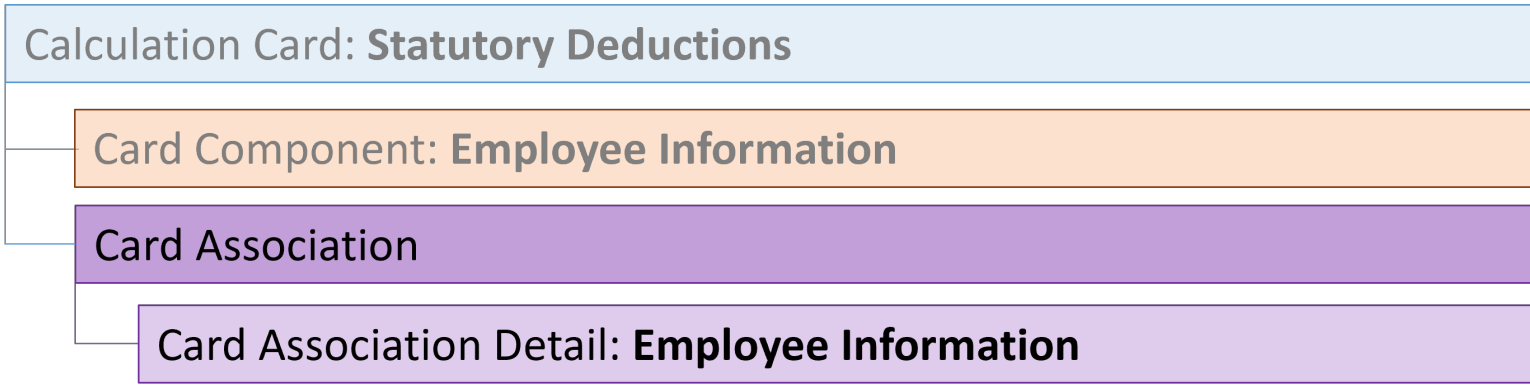
Related Topics

- Guidelines for Loading Statutory Deductions for France

Guidelines for Loading Statutory Deductions Card Associations for France

The card association record associates the Statutory Deductions calculation card with a tax reporting unit.

If you provide the tax reporting unit during the new hire flow, the card association will be created automatically, else it must be created after the Statutory Deductions card is auto-generated. If creating a Statutory Deductions card from scratch you can provide the association details with the Statutory Deductions card.



The associated tax reporting unit is defined in the card association. The card association details records allow the Employee Information card component to be associated with the employee assignments.

Card Association Attributes for Statutory Deductions

The Card Association record type associates a Statutory Deductions card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Statutory Deductions card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Tax Withholding calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Tax Withholding calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Statutory Deductions

The card association details record associates the Employee Information card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Identify the Employee Information card component this association is for.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Example of Creating a New Statutory Deductions card for France

This example creates a complete Statutory Deductions card with associations for assignment E2323.

The CalculationCard.dat file is used to upload Statutory Deductions calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|E2323_SD|FR LDG|Statutory Deductions|2023/01/01|E2323

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|E2323_SD_DET|FR LDG|E2323_SD|2023/01/01|Employee Information

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|FLEX:Deduction
 Developer DF|_EMPLOYEE_SCHEME(Deduction Developer DF=HRX_FR_EMPLOYEE_INFO)|_LOCAL_SCHEME(Deduction
 Developer DF=HRX_FR_EMPLOYEE_INFO)|_EMPLOYEE_PENSION_SCHEME(Deduction Developer DF=HRX_FR_EMPLOYEE_INFO)|
_RESIDENT_ABROAD(Deduction Developer DF=HRX_FR_EMPLOYEE_INFO)|_STATE_PENSION(Deduction Developer
 DF=HRX_FR_EMPLOYEE_INFO)|_COMPLEMENTARY_PENSION(Deduction Developer DF=HRX_FR_EMPLOYEE_INFO)
MERGE|ComponentDetail|VISION|E2323_SD_PREL|FR LDG|E2323_SD_DET|2023/01/01|Employee Information|
HRX_FR_EMPLOYEE_INFO|HRX_FR_EMPLOYEE_INFO|PROFESSIONALIZATION_OVER_45|OTHER_AREAS|NON_MANAGER_APPRENTICE|Y|
N|Y

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|E2323_SD_TRU|FR LDG|E2323_SD|2023/01/01|FR Tax Reporting Unit

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirRepCardId(SourceSystemId)|
DirCardCompId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|E2323_SD_TRU|E2323_SD_TRU|E2323_SD_DET|2023/01/01|E2323
```

Pension and Welfare

Overview of Pensions and Welfare Cards for France

The Pensions and Welfare cards store the employee's situation to pension and welfare contracts.

Contracts between the legal employer and the supplementary pension, welfare, or mutual organization (also known as institutions) are recorded by creating cards at TRU level. Contracts applicable to the employee are identified by creating a card at person level.

Considerations and Prerequisites

You must have created the Pension and Welfare Card at TRU level. Contracts between establishment and pension/ welfare providers (aka institutions) will be stored as a Card at TRU level.

> **Note:** The institutions must be defined as external organization using Locations.

Unlike the Statutory Deductions card, no default card is generated automatically at person level when an employee record is created by the New Hire flow.

Only one card must be created per employee, but several components of the same type (Additional welfare or supplementary pension) can be recorded if more than one contract is applicable for the employee: mutual contract, welfare contract.

It is recommended to have a good understanding of the Pension and Welfare card and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS (France): HR Implementation and Functional Considerations Whitepaper (Doc ID 1542406.1).

Pension and Welfare Card Record Types

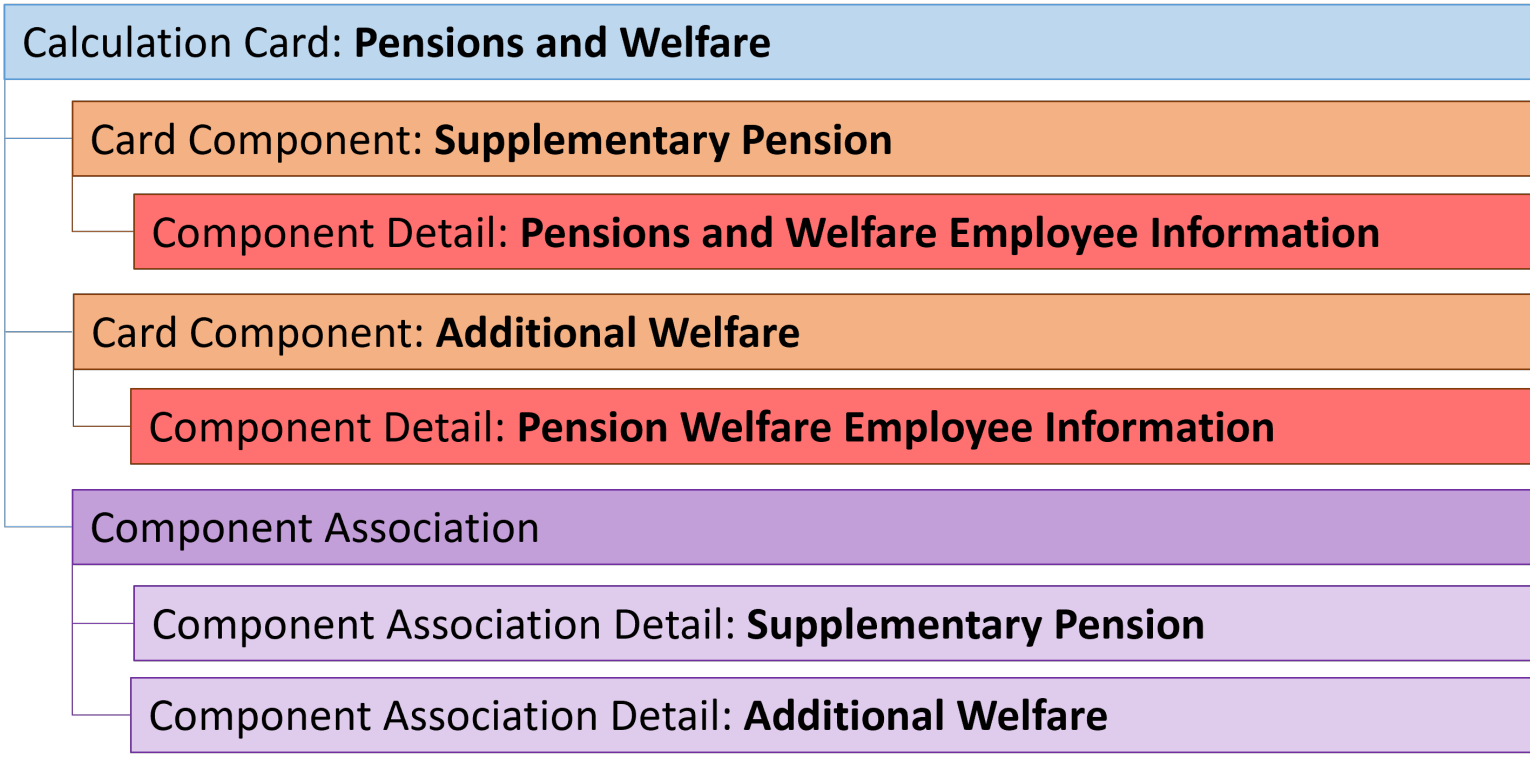
The Pension and Welfare card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Pension and Welfare utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Pension and Welfare Calculation Card Hierarchy

The hierarchy of calculation card components applicable to the Pension and Welfare card are described in this diagram:



Repeat the Supplmentary Pension and Additional Welfare Card Components for each employee contract.

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Pension and Welfare Cards for France

Guidelines for Loading Pension and Welfare Cards for France

You must have one Calculation Card record for every French employee you are maintaining pension and welfare data for.

Even if you are updating an existing Pension and Welfare card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

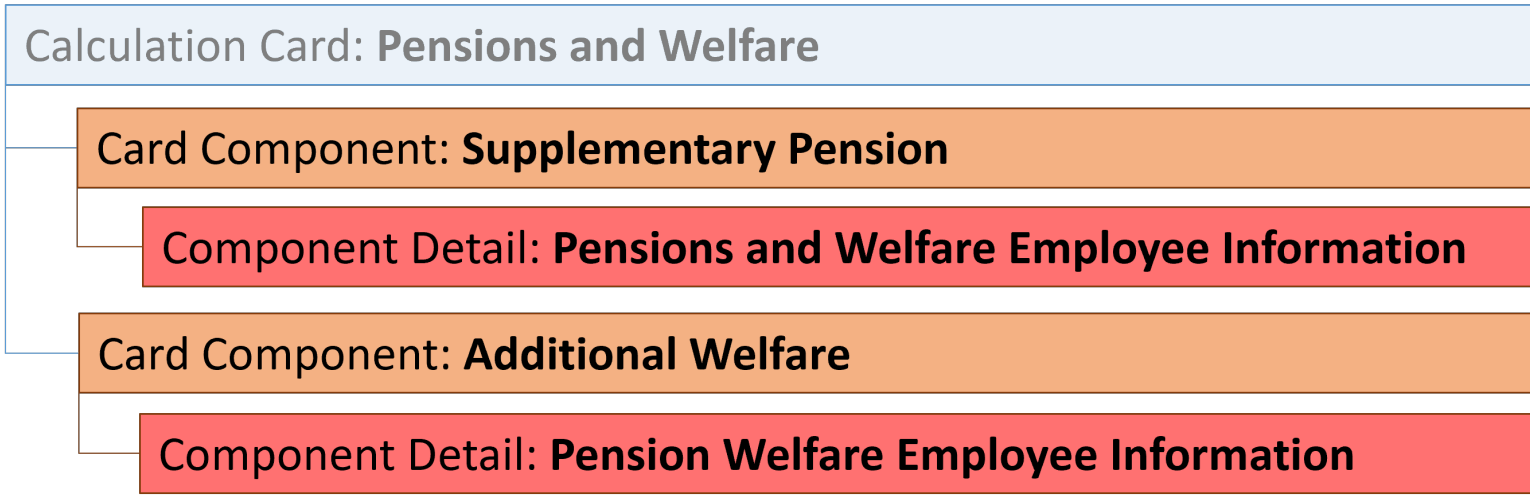
Pension and Welfare Calculation Card Attributes

The Pension and Welfare calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the **Pension and Welfare** calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Pension and Welfare'.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Supplementary Pension and Additional Welfare Card Components

Supply a Supplementary Pension and Additional Welfare card component for each contract for each employee:



Both the he Supplementary Pension and Additional Welfare card components use the Pension Welfare Employee Information flexfield context to capture data. Flexfield segments are uploaded using the Component Detail record type.

Card Component Attributes for Supplementary Pension and Additional Welfare

Both the Supplementary Pension and Additional Welfare card components use these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Supplementary Pension or Additional Welfare card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Pension and Welfare calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Supplementary Pension or Additional Welfare card component. This must be on or after the EffectiveStartDate on the Pension and Welfare calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Supplementary Pension' or 'Additional Welfare'.
ComponentSequence	N/A	Specify '1'.
Context1	N/A	Specify the Declaration Sociale Nominative (DSN) for the employee contract the card component is for. This number must be unique for each employee.

Card Component Detail Attributes for Supplementary Pension and Additional Welfare

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

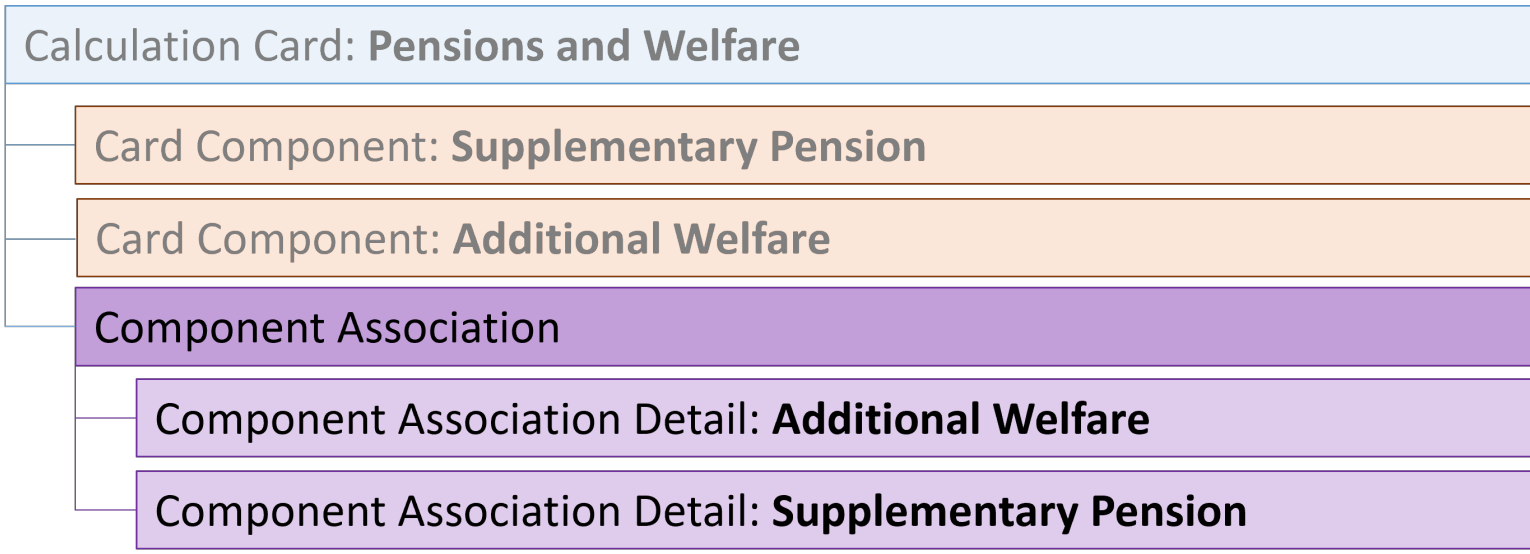
- Pension Welfare Employee Information (HRX_FR_PENSION_WELFARE_INFO)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Supplementary Pension or Additional Welfare card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parent card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_FR_PENSION_WELFARE_INFO'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Pension and Welfare Card Associations

The card association record associates the Pension and Welfare calculation card with a tax reporting unit.



The associated tax reporting unit is defined in the card association. The card association details records allow the Supplementary Pension and Additional Welfare card components to be associated with the employee assignments.

Card Association Attributes for Supplementary Pension and Additional Welfare

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Pension and Welfare card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Pension and Welfare calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Pension and Welfare calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Tax Withholding calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Pension and Welfare

The card association details record associates the Supplementary Pension and Additional Welfare card components with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Identify the Supplementary Pension or Additional Welfare card component this association is for.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Pensions and Welfare Cards for France
- Guidelines for Loading Flexfield Data

Example of Creating a Pension and Welfare Card for France

This example creates a complete Pension and Welfare card with associations for assignment E2323.

Field	Value
Pension Category	222 (Clause 36)
Contract Event	1 (New contract)
Institution Contract Reference	CTRMUTUAL001

The CalculationCard.dat file is used to upload Pension and Welfare calculation cards with HCM Data Loader. Source keys are supplied to uniquely identify each record in the card.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|E2323_PW|FR LDG|Pension and Welfare|2023/01/01|E2323

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|E2323_PW_COMPAW|FR LDG|E2323_PW|2023/01/01|Additional Welfare|MyWelfareContract
MERGE|CardComponent|VISION|E2323_PW_COMPSP|FR LDG|E2323_PW|2023/01/01|Supplementary Pension|
MyPensionContract

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|_PENSION_CATEGORY(Deduction Developer DF=HRX_FR_PENSION_WELFARE_INFO)|
_CONTRACT_EVENT_Display(Deduction Developer DF=HRX_FR_PENSION_WELFARE_INFO)|
_CONTRACT_REFERENCE_Display(Deduction Developer DF=HRX_FR_PENSION_WELFARE_INFO)
MERGE|ComponentDetail|VISION|E2323_PW_FLEXAW|FR LDG|E2323_PW_COMPAW|2023/01/01|Additional Welfare|
HRX_FR_PENSION_WELFARE_INFO|HRX_FR_PENSION_WELFARE_INFO|222|1|CTRMUTUAL001
MERGE|ComponentDetail|VISION|E2323_PW_FLEXSP|FR LDG|E2323_PW_COMPSP|2023/01/01|Supplementary Pension|
HRX_FR_PENSION_WELFARE_INFO|HRX_FR_PENSION_WELFARE_INFO|222|1|CTRMUTUAL001

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|E2323_PW_TRU|FR LDG|E2323_PW|2023/01/01|FR Tax Reporting Unit

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirRepCardId(SourceSystemId)|
DirCardCompId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|E2323_PW_ASAW|E2323_PW_TRU|E2323_PW_COMPAW|2023/01/01|E2323
MERGE|CardAssociationDetail|VISION|E2323_PW_ASSP|E2323_PW_TRU|E2323_PW_COMPSP|2023/01/01|E2323
```

Hardship Card

Overview of Hardship Factor Cards for France

The Hardship Factor card stores information about the risk factors that have been reached for each employee during the year.

Each company must send this information using DSN (Declaration Sociale Nominative) process. This card allows recording the details of all the risk factors for which the employee reaches a certain threshold of hardship.

There are two ways in which the Hardship Factor calculation card can be attached to an employee's record:

- Created or updated manually. In this case the Payroll Manager or Payroll Administrator will use the Manage Calculation Cards UI to create the card and then update the component detail.
- Created or updated in bulk using the HCM Data Loader. Date-effective date is used to track the risk factor per each year. This will trigger an update of the component detail.

Considerations and Prerequisites

Only one card and only one component must be created to record the hardship factor that the employee has been exposed to. Indeed, the customer will initiate the card when one Hardship factor is reached during one year for one employee. The calculation card component will contain the Hardship Factors values and the related year. The following years, user will just need to 'Update' using a new effective date the component detail to change the year and if needed the factors.

Unlike the Statutory Deductions card, no default card is generated automatically at person level when an employee record is created by the New Hire flow

It is recommended to have a good understanding of the Pension and Welfare card and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS (France): HR Implementation and Functional Considerations Whitepaper (Doc ID 1542406.1).

Hardship Factor Card Record Types

The Hardship Factor card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Hardship Factor utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail

Hardship Factor Calculation Card Hierarchy

The hierarchy of calculation card components applicable to the Pension and Welfare card are described in this diagram:

Calculation Card: **Hardship Factor**

Card Component: **Hardship Factors Information**

Card Component Detail: **Hardship Factor Employee Information**

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Pension and Welfare Cards for France

Guidelines for Loading Hardship Factor Cards for France

Supply one Hardship Factor card for every French employee you are maintaining risk factors for.

Even if you're updating an existing calculation card and the calculation card itself isn't being updated, still include the calculation card to group other related data supplied in the file.

Hardship Factor Calculation Card Attributes

The Hardship Factor calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Hardship Factor calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Hardship Factor'.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSeqeuence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		definition exist for the employee. Not required when source keys are used.

Hardship Factors Information Card Components

Supply a single Hardship Factors Information card component for each Hardship Factor card.

Card Component Attributes for Hardship Factors Information

The Hardship Factors Information card components use these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Hardship Factors Information card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Hardship Factor calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card component, typically the employee's start date. This must be after or equal to the EffectiveStartDate on the Hardship Factor calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Hardship Factors Information'.
ComponentSequence	N/A	Specify '1'.

Card Component Detail Attributes for Hardship Factors Information

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield context:

- Hardship Factors Employee Information (ORA_HRX_FR_HARDSHIP_FACTOR_INFO)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Hardship Factors Information card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parent Hardship Factors information card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component. Supply 'Hardship Factors information'.
DirInformationCategory	N/A	The code for the flexfield context, such as 'ORA_HRX_FR_HARDSHIP_FACTOR_INFO'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Hardship Factor Cards for France
- Guidelines for Loading Flexfield Data

Example of Creating a Hardship Factor Card for France

This example creates a complete Pension and Welfare card with associations for assignment E2323.

Field	Value
Hardship Year	2022
Hardship Factor 1	05 (Hyperbaric activities)

The CalculationCard.dat file is used to upload Pension and Welfare calculation cards with HCM Data Loader. Source keys are supplied to uniquely identify each record in the card.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|E2323_HF|FR LDG|Hardship Factor|2022/01/01|E2323

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|E2323_HFI|FR LDG|E2323_HF|2022/01/01|Hardship Factors Information

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|yearOfHardship(Deduction Developer DF=ORA_HRX_FR_HARDSHIP_FACTOR_INFO)|
hardshipFactor1(Deduction Developer DF=ORA_HRX_FR_HARDSHIP_FACTOR_INFO)
MERGE|ComponentDetail|VISION|E2323_HFID|FR LDG|E2323_HFI|2022/01/01|Hardship Factors Information|
ORA_HRX_FR_HARDSHIP_FACTOR_INFO|ORA_HRX_FR_HARDSHIP_FACTOR_INFO|2022|05
```

26 Loading Payroll Localization Data for Netherlands

Statutory Deductions

Overview of Statutory Deductions and Reporting for Netherlands

Statutory Deductions and Reporting cards store all the information required to accurately compute tax contributions for an employee.

It also stores the social insurance contributions of their employer, including standard-rate tax codes and details of any tax discounts to which the employee is eligible, special-rate income, and sector fund details. It includes information used in the statutory wage report and in the digital absence report. One or more Statutory Deductions and Reporting card must be created for each employee. Additional cards need to be created if employees have multiple assignments in different TRUs and/or have different tax rules for each assignment.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, Statutory Deductions and Reporting cards are automatically created when a new employee is entered using the New Hire task with a set of default values, some of which are specified at TRU level.

TRU association and association details for the card are created automatically when the Reporting Establishment and payroll are selected during the hire process. The payroll frequency for the assignment must also match the payroll frequency of the TRU that is defaulted from the reporting establishment

However, there may be cases where this information must be loaded in bulk:

During data migration:

If HCM Data Loader is used to migrate employee records a default Statutory Deductions and Reporting card is automatically created. Default value may need to be updated.

Ongoing bulk updates:

- Bulk loading of new hire data: If you have to do a mass upload of New Hire information, a default Statutory Deductions and Reporting card may be automatically generated (if the new hire records are created through HCM Data Loader). In this case, you need to update the default card with the correct information.

An Employer Deduction Information card must exist for each TRU with which the Statutory Deductions and Reporting card is associated.

You must create element eligibility for the Tax and Social Insurance Calculations element for each Legislative Data Group (LDG) that is used for entering personal calculation cards. Do not check the Automatic entry checkbox while creating element eligibility. It is recommended to have a good understanding of the Statutory Deductions and Reporting card and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS (Netherlands): Payroll Implementation and Functional Considerations (Doc ID 1572711.1).

Statutory Deductions and Reporting Card Record Types

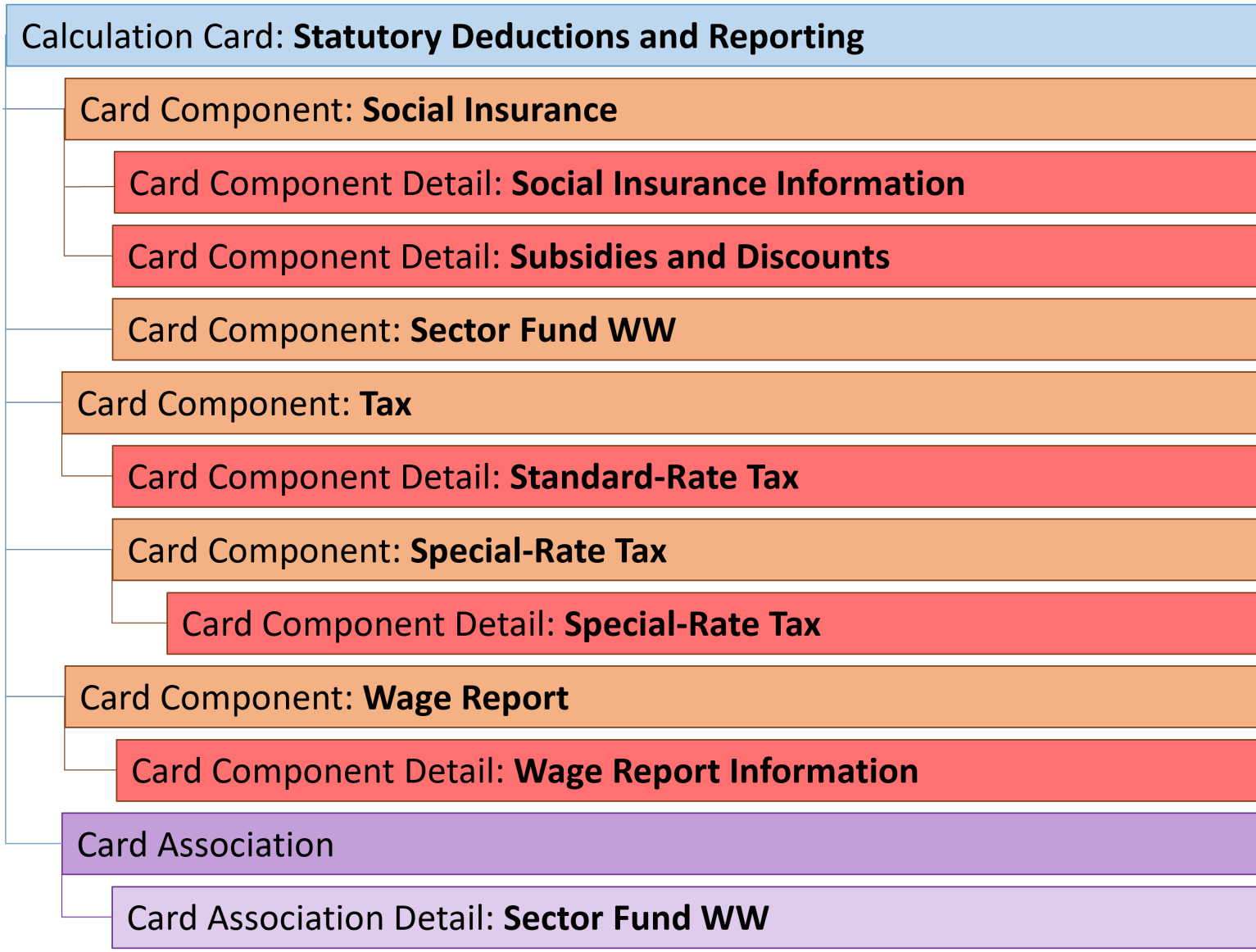
The **Statutory Deductions and Reporting** card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Statutory Deductions and Reporting card uses the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Calculation Value Definition	Allows the creation of value definitions so that overriding values can be specified on the card component.	CalculationValueDefinition
Enterable Calculation Value	For each calculation value definition an overriding value can be specified using the Enterable Calculation Value record type.	EnterableCalculationValue
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Statutory Deductions and Reporting Calculation Card Hierarchy

The hierarchy of calculation card components applicable Statutory Deductions and Reporting card are described in this diagram:



A separate topic follows to describe how to provide data for the calculation card record and each card component.

Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Statutory Deductions and Reporting for Netherlands

Guidelines for Loading Statutory Deductions and Reporting for Netherlands

Supply one (or more) Statutory Deductions and Reporting card record for every Netherlands employee you are maintaining statutory deductions and wage report information for.

Even if you are updating an existing calculation card and the calculation card itself is not being updated, still include the calculation card record to group other related data supplied in the file.

Statutory Deductions and Reporting Calculation Card Attributes

The Statutory Deductions and Reporting calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Statutory Deductions and Reporting calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Statutory Deductions and Reporting'.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
Card Sequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Statutory Deductions and Reporting for Netherlands

Guidelines for Updating Tax for Dutch Statutory Deductions and Reporting

The Tax card component records an employee's standard-rate and special-rate tax information, this includes their tax code, income code, and previous year's special-rate income.

Calculation Card: **Statutory Deductions and Reporting**

Card Component: **Tax**

Card Component Detail: **Standard-Rate Tax**

The Tax card component uses a single flexfield context which is loaded using the Component Detail record type.

Card Component Attributes for Tax

The Tax card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Tax card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions and Reporting calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Tax card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Statutory Deductions and Reporting calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Tax'.
ComponentSequence	N/A	The number to uniquely identify the Tax card component when multiple PAYE card components exist. Not required when the source key is used to identify the card component.
Context1	N/A	The name of the TRU with which the card must be associated.

Component Detail Attributes for Tax

The Tax card component uses a single flexfield context. In addition to the attributes defined here, include the necessary flexfield segment attribute values for this flexfield context:

- Standard-Rate Tax (HRX_NL_STD_TAX)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **Tax** card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the **Tax** card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_NL_STD_TAX'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Example of Updating Tax

This example updates the existing Tax card component for employee assignment E1212 on the 1st January 2021. User keys are provided on the assumption that the Statutory Deductions and Reporting card was auto generated and therefore the source key values are unknown.

The following flexfield segments are updated for the Standard-Rate Tax context:

Field	Value
Income Code	AOW allowance (22)
Tax Type	Social insurance only (3)
Tax Table	Green table (2)
Payroll Period	2 - Monthly (2)

The CalculationCard.dat file is used to upload the Tax card component information with HCM Data Loader.

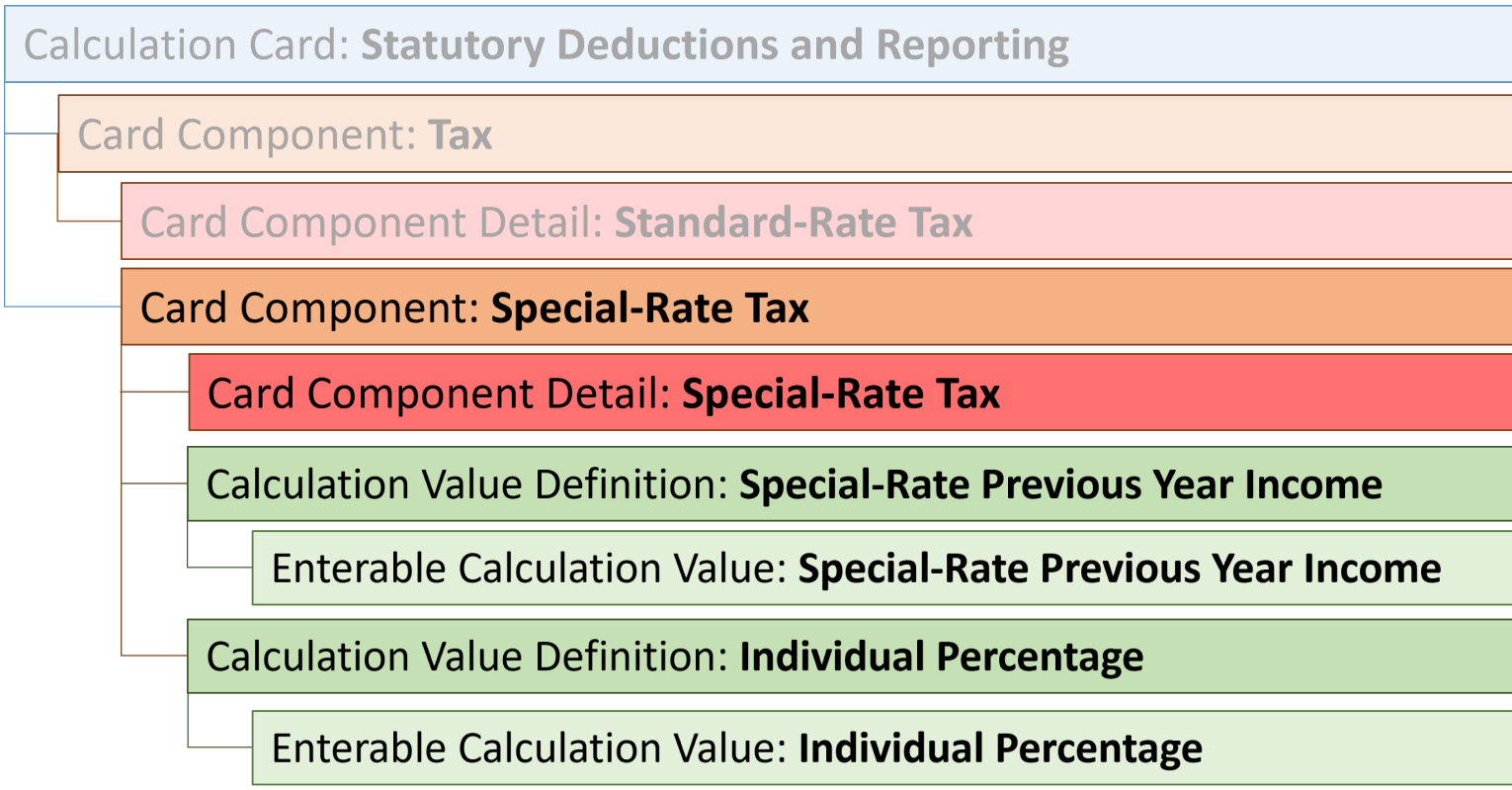
```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|NL LDG|Statutory Deductions and Reporting|2020/01/01|E1212|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
DirCardCompDefName|CardSequence|ComponentSequence
MERGE|CardComponent|NL LDG|Statutory Deductions and Reporting|2020/01/01|E1212|Tax|1|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
DirCardCompDefName|CardSequence|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
_INCOME_CODE(Deduction Developer DF=HRX_NL_STD_TAX)|_TAX_TYPE(Deduction Developer DF=HRX_NL_STD_TAX)|
_TAX_TABLE(Deduction Developer DF=HRX_NL_STD_TAX)|_PAYROLL_PERIOD(Deduction Developer DF=HRX_NL_STD_TAX)
MERGE|ComponentDetail|NL LDG|Statutory Deductions and Reporting|2021/01/01|E1212|Tax|1|1|HRX_NL_STD_TAX|
HRX_NL_STD_TAX|22|3|2|2
```

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Statutory Deductions and Reporting for Netherlands
- Guidelines for Loading Flexfield Data

Guidelines for Updating Special-Rate Tax for Dutch Statutory Deductions and Reporting

The Special-Rate Tax card component records all additional tax relevant information, including the previous year's special-rate income on which the special-rate percentage is based and an individual special-rate tax rate, if appropriate.



The Special-Rate Tax card component uses a single flexfield context which is supplied using the the Component Detail record. There are two value definitions supported which are loaded using the Calculation Value Definition and Enterable Calculation Value record types.

The Special-Rate Tax card component references the Tax card component as its parent.

Card Component Attributes for Tax

The Special-Rate Tax card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Special-Rate Tax card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions and Reporting calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
ParentDirCardCompId(Sour ceSystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	Identify the Tax card component for the employee. You can either provide this attribute with the SourceSystemId value supplied for the Tax card component or provide the user key attributes.
EffectiveStartDate	N/A	The start date of the Special-Rate Taxcard component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Statutory Deductions and Reporting calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Special-Rate Tax'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components with the same DirCardCompDefName exists. **Note:** Not required when source keys are used.
Context1	N/A	The name of the TRU with which the card must be associated.

Component Detail Attributes for Special-Rate Tax

The Special-Rate Tax card component uses a single flexfield context. In addition to the attributes defined here, include the necessary flexfield segment attribute values for this flexfield context:

- Special-Rate Tax (HRX_NL_SPECIAL_RATE_TAX)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute Alternative User Key Attributes Functional Description

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Special-Rate Tax card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Special-Rate Tax card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_NL_SPECIAL_RATE_TAX'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Calculation Value Definition Attributes

The calculation value definition allows the creation of a rate definition so that overriding rates can be provided on the Statutory Deductions and Reporting calculation card.

The discriminator CalculationValueDefinition is used to load calculation value definition records using HCM Data Loader. The following CalculationValueDefinition attributes are commonly provided when loading a new Statutory Deductions and Reporting card.

The Special-Rate Tax card component utilizes two value definitions to supply overrides values; Special-Rate Previous Year Income and Individual Percentage.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Special-Rate Tax card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent Special-Rate Tax card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent Special-Rate Tax card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. Supply either of the following: • Special-Rate Previous Year Income • Individual Percentage

Enterable Calculation Value Attributes

The enterable calculation value allows the creation of the actual value of the calculation value definition created above.

The discriminator EnterableCalculationValue is used to load enterable calculation value records using HCM Data Loader. The following EnterableCalculationValue attributes are commonly provided when loading a new Statutory Deductions and Reporting card.

HCM Data Loader Attribute Alternative User Key Attributes Functional Description:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition.

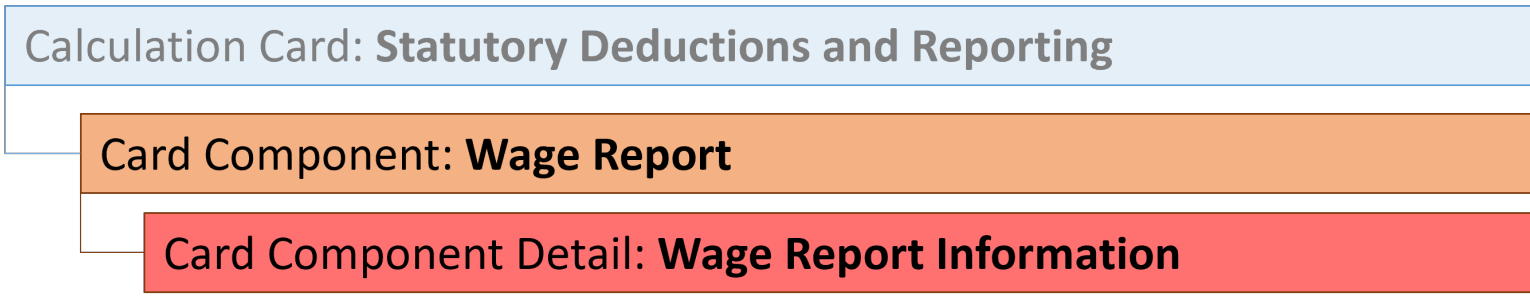
HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record. Supply the value of the rate definition for the Special-Rate Previous Year Income or Individual Percentage Approved Date override.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Statutory Deductions and Reporting for Netherlands
- Guidelines for Loading Flexfield Data

Guidelines for Updating Wage Report for Dutch Statutory Deductions and Reporting

The Wage Report card component records information for statutory wage report submissions to the tax authority.



The Wage Report card component uses a single flexfield context which is loaded using the Component Detail record type.

Card Component Attributes for Tax Wage Report

The Wage Report card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Wage Report card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions and Reporting calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Wage Report card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Statutory Deductions and Reporting calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Wage Report'.
ComponentSequence	N/A	The number to uniquely identify the Wage Report card component when multiple PAYE card components exist. Not required when the source key is used to identify the card component.

Component Detail Attributes for Wage Report

The Wage Report card component uses a single flexfield context. In addition to the attributes defined here, include the necessary flexfield segment attribute values for this flexfield context:

- Wage Report Information (HRX_NL_WAGE_REPORT_INFO)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **Wage Report** card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the **Wage Report** card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_NL_WAGE_REPORT_INFO'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

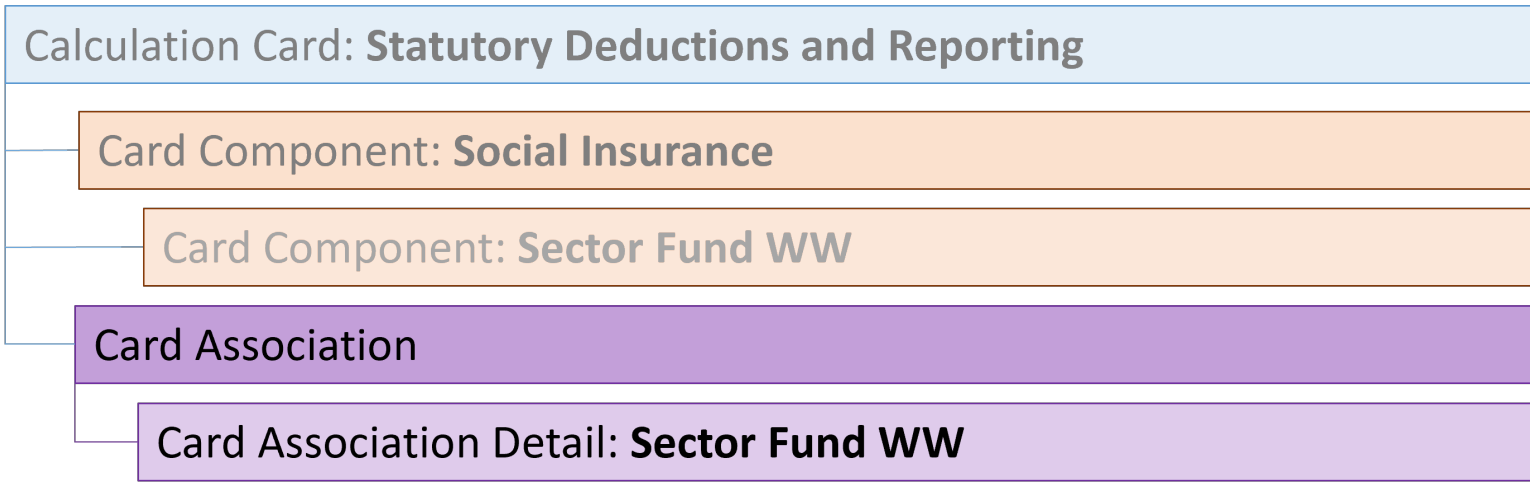
Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Statutory Deductions and Reporting for Netherlands
- Guidelines for Loading Flexfield Data

Guidelines for Loading Dutch Statutory Deductions and Reporting Card Associations

The card association record associates the Statutory Deductions calculation card with a tax reporting unit.

If you provide the tax reporting unit during the new hire flow, the card association will be created automatically, else it must be created after the Statutory Deductions and Reporting card is auto-generated. If creating a Statutory Deductions and Reporting card from scratch, you can provide the association details with the Statutory Deductions and Reporting card.



Card Association Attributes for Statutory Deductions and Reporting

The Card Association record type associates a Statutory Deductions and Reporting card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Statutory Deductions and Reporting card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions and Reporting calculation card should be identified by using the same key type used to identify the calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Tax Withholding calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Tax Withholding calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Statutory Deductions and Reporting

The card association details record associates the NI and PAYE card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName,	Identify the Sector Fund WW card component this association is for.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, DirCardCompDefName	
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Example of Updating the Dutch Card Association Detail

This example updates the association detail of the Statutory Deductions and Reporting card belonging to assignment E1212.

User keys are provided to reference the existing Statutory Deductions and Reporting card on the assumption it was auto generated and therefore the source keys are unknown.

The CalculationCard.dat file is used to upload the card association and to add a new association detail with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|NL LDG |Statutory Deductions and Reporting|2021/01/01|E1212|1

METADATA|CardAssociation|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|TaxReportingUnitName
MERGE|CardAssociation|NL LDG|Statutory Deductions and Reporting|2021/01/01|E1212|1|NL Tax Reporting Unit

METADATA|CardAssociationDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|
AssignmentNumber|CardSequence|TaxReportingUnitName|DirCardCompDefName|ComponentSequence|
AssociationAssignmentNumber
MERGE|CardAssociationDetail|NL LDG|Statutory Deductions and Reporting|2021/01/01|E1212|1|NL Tax Reporting
 Unit|Sector Fund WW|1|E1212
```

Related Topics

Example of Creating a Statutory Deductions and Reporting Card for Netherlands

In this example the complete Statutory Deductions and Reporting card with associations is created for employee assignment E1414.

The CalculationCard.dat file is used to upload Statutory Deductions and Reporting Card calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|HRC_SQLLOADER|E1414_SD|NL LDG|Statutory Deductions and Reporting|2023/01/01|E1414

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1|ParentDirCardCompId(SourceSystemId)
MERGE|CardComponent|HRC_SQLLOADER|E1414_SD_TX|NL LDG|E1414_SD|2023/01/01|Tax|NL Tax Reporting Unit|
MERGE|CardComponent|HRC_SQLLOADER|E1414_SD_STX|NL LDG|E1414_SD|2023/01/01|Special-Rate Tax||E1414_SD_TX
MERGE|CardComponent|HRC_SQLLOADER|E1414_SD_SI|NL LDG|E1414_SD|2023/01/01|Social Insurance||
MERGE|CardComponent|HRC_SQLLOADER|E1414_SD_SF|NL LDG|E1414_SD|2023/01/01|Sector Fund WW|1 - 01|E1414_SD_SI
```

```
METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|FLEX:Deduction
 Developer DF|_INCOME_CODE(Deduction Developer DF=HRX_NL_STD_TAX)|_TAX_TYPE(Deduction Developer
 DF=HRX_NL_STD_TAX)|_TAX_TABLE(Deduction Developer DF=HRX_NL_STD_TAX)|_PAYROLL_PERIOD(Deduction
 Developer DF=HRX_NL_STD_TAX)|_ZVW_INSURED(Deduction Developer DF=HRX_NL_SOCIAL_INSURANCE_INFO)|
_WAO_DISCOUNT_ELIGIBLE(Deduction Developer DF=HRX_NL_SOCIAL_INSURANCE_INFO)|oraHrxNlNoRiskZwPol(Deduction
 Developer DF=HRX_NL_SOCIAL_INSURANCE_INFO)
MERGE|ComponentDetail|HRC_SQLLOADER|E1414_SD_TAX|NL LDG|E1414_SD_TX|2023/01/01|Tax|HRX_NL_STD_TAX|
HRX_NL_STD_TAX|22|3|2|2|||
MERGE|ComponentDetail|HRC_SQLLOADER|E1414_SD_PREL|NL LDG|E1414_SD_SI|2023/01/01|Social Insurance|
HRX_NL_SOCIAL_INSURANCE_INFO|HRX_NL_SOCIAL_INSURANCE_INFO|||||M|Y|Y

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|HRC_SQLLOADER|E1414_SD_TRU|NL LDG|E1414_SD|2023/01/01|NL Tax Reporting Unit

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirRepCardId(SourceSystemId)|
DirCardCompId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|HRC_SQLLOADER|E1414_SD_ASS|E1414_SD_TRU|E1414_SD_SF|2023/01/01|E1414
```

27 Loading Payroll Localization Data for Mexico

Employee Tax Card

Overview of Employee Tax Card for Mexico

The Employee Tax Card calculation card stores all information required to accurately compute employee state tax, employee ISR tax, contributions for social insurance and loans (Employee INFONAVIT).

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, Employee Tax Card calculation cards are automatically created when a new employee is entered using the New Hire task with a set of default values specified at the Payroll Statutory Unit (PSU) or Tax Reporting Unit (TRU) level.

TRU association and association details for the employee are created automatically when the TRU is selected during the hire process. There may be cases where this information needs to be loaded in bulk:

During Data Migration:

Employee Tax Card information must be uploaded into Oracle Fusion Payroll, to ensure that contributions and taxes are calculated correctly.

If HCM Data Loader is used to migrate employee records, a default Employee Tax Card is automatically created. In some cases, the default won't reflect the employee's actual calculation information, and therefore the card must be updated.

Ongoing Bulk Updates:

- When you bulk load New Hire information, a default Employee Tax Card may be automatically generated (if the new hire records are created through HCM Data Loader or the interface with Taleo). In this case, you need to update the default card with the correct information and create the association for the TRU.

It is recommended to have a good understanding of the Employee Tax Card calculationcard and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS (Mexico): HR Implementation and Use (Doc ID 2175160.1) and Oracle Fusion Capital Management for Mexico: Federal Income Tax (ISR) (Doc ID 2541656.1).

Employee Tax Card Record Types

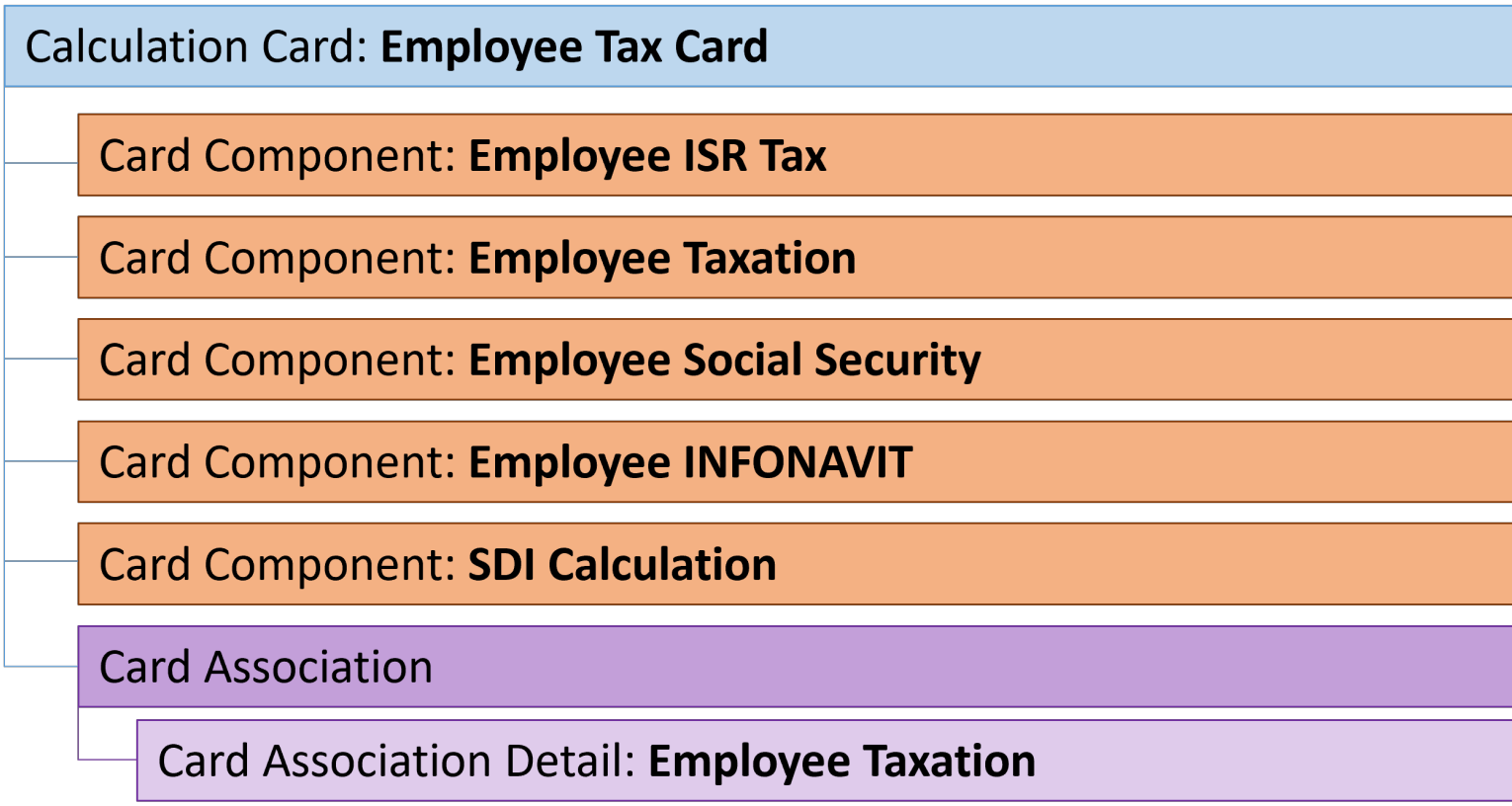
The Employee Tax Card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Employee Tax Card utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Calculation Value Definition	Allows the creation of value definitions so that overriding values can be specified on the card component. Details of the specific value definitions are provided in the following sections.	CalculationValueDefinition
Enterable Calculation Value	Used to specify an overriding value for each calculation value definition.	EnterableCalculationValue
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Employee Tax Card Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Employee Tax Card are described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Employee Tax Cards for Mexico
- Guidelines for Loading Flexfield Data

Guidelines for Loading Employee Tax Cards for Mexico

You must have one Calculation Card record for every Mexico employee you are maintaining taxes, social insurance, and loans data for.

Even if you are updating an existing Employee Tax Card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Employee Tax Card Calculation Card Attributes

The Employee Tax Card calculation card uses these attributes:

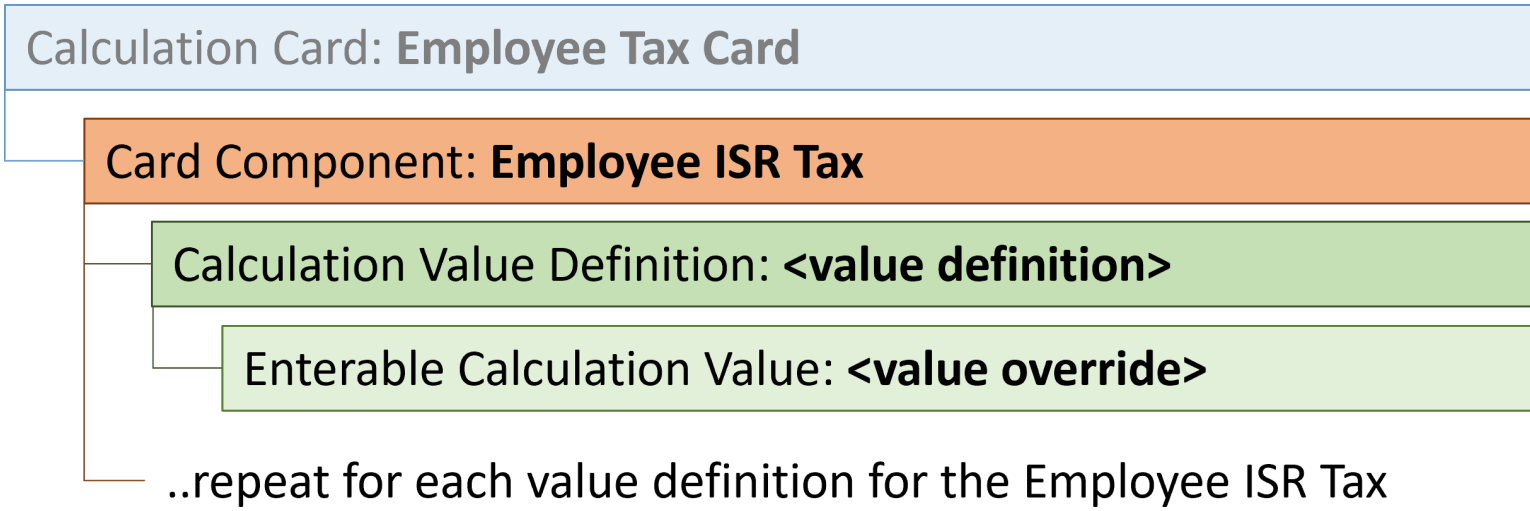
HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Employee Tax Card calculation card. For new calculation card supply the source key attributes. You can also

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Employee Tax Card'.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Employee ISR Tax Card Component

The Employee ISR Tax card component captures tax related information which allows overrides for the Employee ISR Tax Information value definition group. This card component is automatically generated for you if the Employee Tax Card is automatically generated.

If you need to update the defaulted tax information, provide an Employee ISR Tax card component and the value definitions and value overrides.



Card Component Attributes for Employee ISR Tax

The Employee ISR Tax card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee ISR Tax card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Tax Card calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Employee ISR Tax card component, typically the employee's start date. This must be on or after the EffectiveStartDate on the Employee Tax Card calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Employee ISR Tax'.
ComponentSequence	N/A	Specify '1'.

Employee ISR Tax Value Definitions

The Employee ISR Tax card component uses value definitions to supply override values:

Value Definition Name	Functional Description
Exclude From Annual Adjustments	Employee's exemption to ISR Annual Adjustments (overrides the value stored in the Calculation Value Definition). Provides the option to exclude an employee from the annual tax adjustment process. The default value for this field is No. If you want to exclude the employee from the annual tax adjustment, select Yes.
ISR Severance Rule	Determines which Severance ISR tax calculation method to use. The default is Subject Earnings Greater Than or Equal to Monthly Salary. This means that the Severance tax calculation method is used if the Severance ISR Subject portion is greater than or equal to the Monthly Salary.

Value Definition Name	Functional Description
	If you select None, the severance ISR calculation method is used with no comparison and regardless of the severance amount. If you choose Total Earnings Greater Than or Equal to Monthly Salary, the severance tax calculation method is used if the total severance payment (subject plus exempt portions) is greater than or equal to the Monthly Salary.
ISR Subsidy Adjustment on Regular Run Type	For each payroll run, the standard ISR calculation adjusts the amount of subsidy for employment to prevent exceeding the monthly cap. This adjustment compares the month-to-date (MTD) subject earnings balance with the prorated monthly subsidy for employment rates. This override determines if the ISR Subsidy for Employment is adjusted during a Regular payroll run type. Subsidy for employment is calculated (adjusted or not) based on the rates determined by the existing ISR Subsidy Proration on Payroll Run Type override. The default is yes, whereby the Subsidy for employment is adjusted based on the MTD ISR subject earnings balance. If no is selected, the Subsidy for employment is calculated with no adjustment considering the current processing pay period ISR subject earnings (Run balance).
ISR Subsidy Proration on Payroll Run Type	This override determines if the ISR Subsidy for Employment is calculated based on the periods in the month and on the proration of the monthly Subsidy for Employment rates. The Subsidy for Employment calculation basis will be the MTD ISR subject earnings balance in all scenarios except if the ISR Subsidy for employment Adjustment on Regular payroll run override is set to No. In this case, the basis will be the Run balance. The payroll run type(s) excluded by this value definition will use a Subsidy for Employment rates factor of one (no proration). This means that the full monthly rates are used to calculate the Subsidy for Employment. The default is Regular and Monthly Tax Adjustment (Default). This means that the Subsidy for Employment Proration is applied to both payroll run types: Monthly Tax Adjustment and Regular. For None, no Subsidy for Employment proration (factor 1) is applied to both payroll run types, Monthly Tax Adjustment and Regular. If you select Regular, proration is applied only to the Regular run type. The Monthly Tax Adjustment run type will use a Subsidy for Employment proration factor of one. If you select the Monthly Tax Adjustment, proration is applied only to the Monthly Tax Adjustment run type. Regular run type will use a Subsidy for Employment proration factor of one.
ISR Tax Calculation	This override determines if the ISR tax is calculated. The default value is 'Yes'. **Note:** If this override is set to 'No', it takes precedence over all other Employee ISR Tax component overrides.
ISR Tax Proration on Monthly Tax Adjustment Run Type	The standard Monthly Tax Adjustment ISR tax calculation procedure calculates the tax rates based on a factor derived from the periods the employee has worked in the month. This override applies only to the Monthly Tax Adjustment payroll run type and determines to calculate the tax based on the periods in the month. The tax calculation basis will be the month-to-date (MTD) ISR subject earnings balance. The default is yes. This means that the ISR tax calculation determines the tax rates based on a factor derived from the periods the employee has worked in the month. If you select No, the ISR Tax rates are applied with a factor of one (no proration), regardless of the period when the Monthly Tax Adjustment run type is processed nor the employee hire date.

Value Definition Name	Functional Description
Pay Subsidy for Employment on Payroll Run Type	When an employee has two or more employers, they must select which employer will pay the subsidy for employment in the ISR tax calculation. The options to determine when the subsidy is calculated include: • Monthly Adjustment The subsidy for employment is calculated only for the monthly tax adjustment run type. • None The subsidy for employment is never calculated. • Regular The subsidy for employment is calculated only for the regular run type. • Regular and Monthly Tax Adjustment (default) The subsidy for employment is calculated for both regular and monthly tax adjustment run types.
State Tax Exempt	Employee's exemption to State Tax (overrides the value stored in the Calculation Value Definition). Some companies or government agencies aren't required to pay the employer state tax. Select 'Yes', to exclude this employee from the state tax calculation. The default value for this field is 'No'.

Calculation Value Definition Attributes for Employee ISR Tax

The Calculation Value Definition record type specifies the name of the value definition you are supplying an override value for.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Employee ISR Tax card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The start date of the parent Employee ISR Tax card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent Employee ISR Tax card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent Employee ISR Tax card component and a CalculationCard record for the owning Employee Tax Card.

Enterable Calculation Value Attributes for Employee ISR Tax

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		Unlike other calculation cards, if you supply a value definition but have no value for it supply '-999999999' to indicate a null value.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Employee Taxation Card Component

The Employee Taxation card component captures tax information, which is needed to compute Mexico taxes. This card component is automatically generated for you if the Employee Tax Card is automatically generated. If you need to update the defaulted tax information, provide an Employee Taxation card component and the value definitions and value overrides.

Calculation Card: **Employee Tax Card**

Card Component: **Employee Taxation**

Card Component Attributes for Employee Taxation

The Employee Taxation card component uses these attributes:

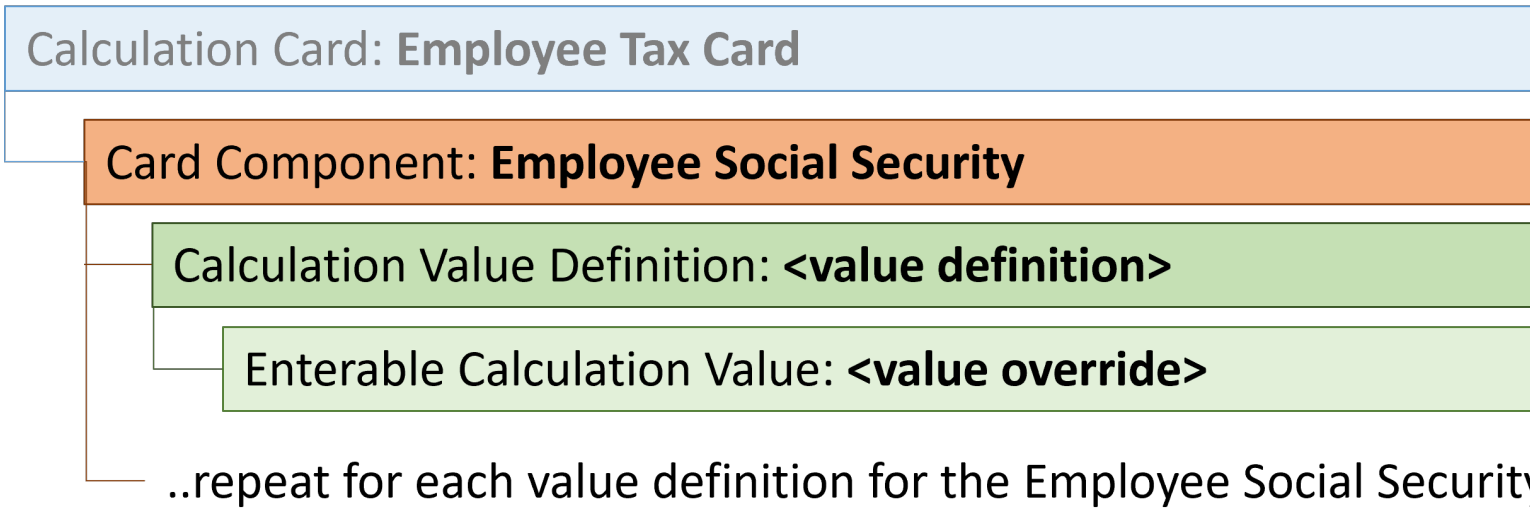
HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee Taxation card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Tax Card calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Employee Taxation card component, typically the employee's start date. This must be on or after the EffectiveStartDate on the Employee Tax Card calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Employee Taxation'.
ComponentSequence	N/A	Specify '1'.

Employee Social Security Card Component

The Employee Social Security card component captures social insurance related information which is needed to calculate the employee social security and retirement contributions. This card component is automatically generated for you if the Employee Tax Card is automatically generated.

If you need to update the defaulted social insurance information, provide an Employee Social Security card component and the applicable value definitions and value overrides.



Card Component Attributes for Employee Social Security

The Employee Social Insurance card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee Social Security card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Tax Card calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Employee Social Security card component, typically the employee's start date. This must be on or after the EffectiveStartDate on the Employee Tax Card calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Employee Social Security'.
ComponentSequence	N/A	Specify '1'.

Employee Social Security Value Definitions

The Employee Social Security card component uses value definitions to supply override values.

Value Definition Name	Functional Description
Economic Zone	To override the Minimum Wage Economic Zone of the TRU the employee is associated with, select either Zone A or Zone B.
Additional Quota Exempt	Determines if the employee is exempt from having the Additional Quota calculated. The default value for this field is No. If you want to exclude the employee from Additional Quota calculation, select 'Yes'.
Additional Quota Percentage	Enter the percentage of additional quota for this employee if it's required to be different to the default.
Benefits in Cash Exempt	Determines if the employee is exempt from having Benefits in Cash calculated. The default value for this field is 'No'. If you want to exclude the employee from Benefits in Cash, select 'Yes'.
Benefits in Cash Percentage	Enter the percentage of benefits in cash for this employee if it's required to be different to the default.
Day Care Centers Exempt	Determines if the employee is exempt from having Day Care Centers calculated. The default value for this field is No. If you want to exclude the employee from Day Care Centers, select 'Yes'.
Day Care Centers Percentage	Enter the percentage of day care centers for this employee if it's required to be different to the default.
Disability and Death Exempt	Determines if the employee is exempt from having Disability and Death calculated. The default value for this field is 'No'. If you want to exclude the employee from Disability and Death, select 'Yes'.

Value Definition Name	Functional Description
Disability and Death Percentage	Enter the percentage of disability and death for this employee if it's required to be different to the default.
Fixed Quota Exempt	Determines if the employee is exempt from having Fixed Quota calculated. The default value for this field is No. If you want to exclude the employee from Fixed Quota, select 'Yes'.
Fixed Quota Percentage	Enter the percentage of fixed quota for this employee if it's required to be different to the default.
Mexico TRU Transfer Leaving Reason	Records the social security leaving reason when an employee transfers from one tax reporting unit to another. Validated by lookup type ORA_HRX_MX_SS_LEAVE_REASON.
Pensions and Beneficiaries Exempt	Determines if the employee is exempt from having Pensions and Beneficiaries calculated. The default value for this field is No. If you want to exclude the employee from Pensions and Beneficiaries, select Yes.
Pensions and Beneficiaries Percentage	Enter the percentage of pensions and beneficiaries for this employee if it's required to be different to the default.
Retirement Exempt	Determines if the employee is exempt from having Retirement calculated. The default value for this field is No. If you want to exclude the employee from Retirement, select 'Yes'.
Retirement Percentage	Enter the percentage of retirement for this employee if it's required to be different to the default.
Social Security Exemption	Determines if the employee is exempt from having social security quota liabilities calculated. The default value for this field is No. If you want to exclude the employee from having social security quota liabilities calculated, select 'Yes'.
Suspension in Age Outpost and Oldness Exempt	Determines if the employee is exempt from having Suspension in Age of Outpost and Oldness calculated. The default value for this field is No. If you want to exclude the employee from Suspension in Age Outpost and Oldness, select 'Yes'.
Suspension in Age Outpost and Oldness Percentage	Enter the percentage of Suspension in Age of Outpost and Oldness for this employee if it's required to be different to the default.
Tax Reporting Unit for Social Security	Select the tax reporting unit to be used for reporting this employee's social security quotas if it's different to the calculation card Association's tax reporting unit.

Calculation Value Definition Attributes for Employee Social Security

The Calculation Value Definition record type specifies the name of the value definition you are supplying an override value for.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Employee Social Security card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent Employee Social Security card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent Employee Social Security card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent Employee Social Security card component and a CalculationCard record for the owning Employee Tax Card.

Enterable Calculation Value Attributes for Employee Social Security

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
ValueDefnId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record. Unlike other calculation cards, if you supply a value definition but have no value for it supply '-999999999' to indicate a null value.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefinition record supplied.

Example of Updating Employee Tax Cards for Mexico

This example updates an auto-generated Employee Tax Card card to override Employee ISR Tax and Employee Social Security values.

User keys are used to identify the card and card component as the source keys may not be known for auto-generated cards. Use the CalculationCard.dat file to update Employee Tax Card card component information with HCM Data Loader.

The Employee ISR Tax card component is updated, overriding this value:

Value Definition	Value
Pay Subsidy For Employment	No

The Employee Social Security card component is updated, overriding these values:

Value Definition	Value
Additional Quota Exempt	No
Additional Quota Percentage	1

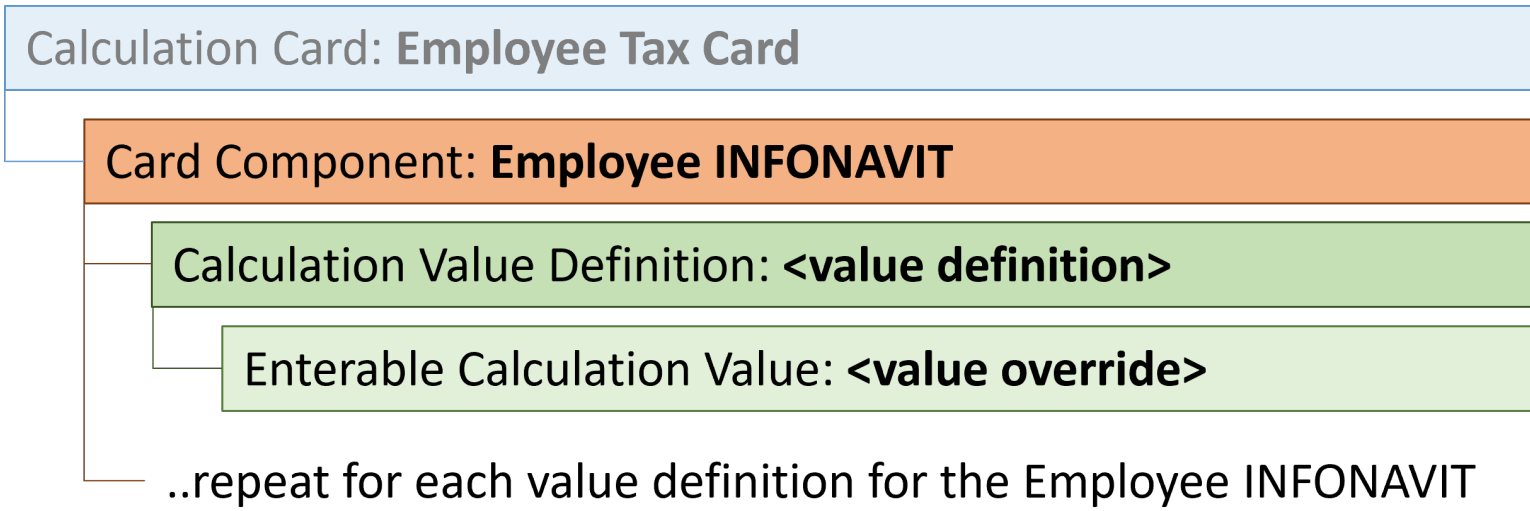
```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|MX LDG|Employee Tax Card|2023/01/01|E1515|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
DirCardCompDefName|CardSequence|ComponentSequence
MERGE|CardComponent|MX LDG|Employee Tax Card|2023/01/01|E1515|Employee ISR Tax|1|1
MERGE|CardComponent|MX LDG|Employee Tax Card|2023/01/01|E1515|Employee Taxation|1|1
MERGE|CardComponent|MX LDG|Employee Tax Card|2023/01/01|E1515|Employee Social Security|1|1

METADATA|CalculationValueDefinition|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|CardSequence|ComponentSequence|ValueDefinitionName
MERGE|CalculationValueDefinition|MX LDG|Employee Tax Card|2023/01/01|E1515|Employee ISR Tax|1|1|Pay Subsidy
 For Employment
MERGE|CalculationValueDefinition|MX LDG|Employee Tax Card|2023/01/01|E1515|Employee Social Security|1|1|
Additional Quota Exempt
MERGE|CalculationValueDefinition|MX LDG|Employee Tax Card|2023/01/01|E1515|Employee Social Security|1|1|
Additional Quota Percentage

METADATA|EnterableCalculationValue|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|CardSequence|ComponentSequence|ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|MX LDG|Employee Tax Card|2023/01/01|E1515|Employee ISR Tax|1|1|Pay Subsidy
 For Employment|No
MERGE|EnterableCalculationValue|MX LDG|Employee Tax Card|2023/01/01|E1515|Employee Social Security|1|1|
Additional Quota Exempt|No
MERGE|EnterableCalculationValue|MX LDG|Employee Tax Card|2023/01/01|E1515|Employee Social Security|1|1|
Additional Quota Percentage|0.1
```

Guidelines for Loading Employee INFONAVIT for the Mexico Employee Tax Card

These are the guidelines for loading the INFONAVIT card component.

Employee INFONAVIT Card Component

The Employee INFONAVIT card component captures all housing loan related information which is needed to calculate employee INFONAVIT loan deductions.



Card Component Attributes for Employee INFONAVIT

The Employee INFONAVIT card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee INFONAVIT card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Tax Card calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Employee INFONAVIT card component, typically the employee's start date. This must be on or after the EffectiveStartDate on the Employee Tax Card calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Employee INFONAVIT'.
ComponentSequence	N/A	Specify '1'.

Employee INFONAVIT Value Definitions

The Employee INFONAVIT card component uses value definitions to supply override values:

Value Definition Name	Functional Description
Credit Number	Employee's INFONAVIT credit number.
Credit Start Date	The start date of the credit.
Deduction Table	Specify Yes/No
Discount Type	The type of discount for the credit, such as percentage. Validated by lookup type ORA_HRX_MX_INFONAVIT_DISCOUNT.
Discount Value	The value of the discount.

Value Definition Name	Functional Description
Mexico SUA INFONAVIT Transaction	The SUA INFONAVIT transaction type. Validated by the lookup type ORA_HRX_MX_SUA_INFONAVIT_TRANS.
Use Integrated Daily Wage Cap	Specify Yes/No

Calculation Value Definition Attributes for Employee INFONAVIT

The Calculation Value Definition record type specifies the name of the value definition you are supplying an override value for.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Employee INFORNAVIT card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent Employee INFONAVIT card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent Employee INFONAVIT card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

Enterable Calculation Value Attributes for Employee INFONAVIT

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent Employee INFONAVIT card component and a CalculationCard record for the owning Employee Tax Card.

HCM Data Loader Attribute Alternative User Key Attributes Functional Description:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record. Unlike other calculation cards, if you supply a value definition but have no value for it supply '-999999999' to indicate a null value.

Example of Loading Employee INFONAVIT

This example creates the Employee INFONAVIT component for the automatically generated Employee Tax Card for employee assignment E234116 on 1-Jan-23.

The calculation card is identified using user keys as the source key may be unknown for an auto-generated card. Source keys are used to identify the new card component, value definition and override value records.

The CalculationCard.dat file is used to upload the Employee Tax Card with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|DirCardDefinitionName|LegislativeDataGroupName|AssignmentNumber|
EffectiveStartDate|CardSequence
MERGE|CalculationCard|VISION|Employee Tax Card|MX LDG|E234116|2023/01/01|1

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|AssignmentNumber|
EffectiveStartDate|DirCardCompDefName|DirCardDefinitionName|CardSequence
MERGE|CardComponent|VISION|ETC_234116_INF|MX LDG|E234116|2023/01/01|Employee INFONAVIT|Employee Tax Card|1
```

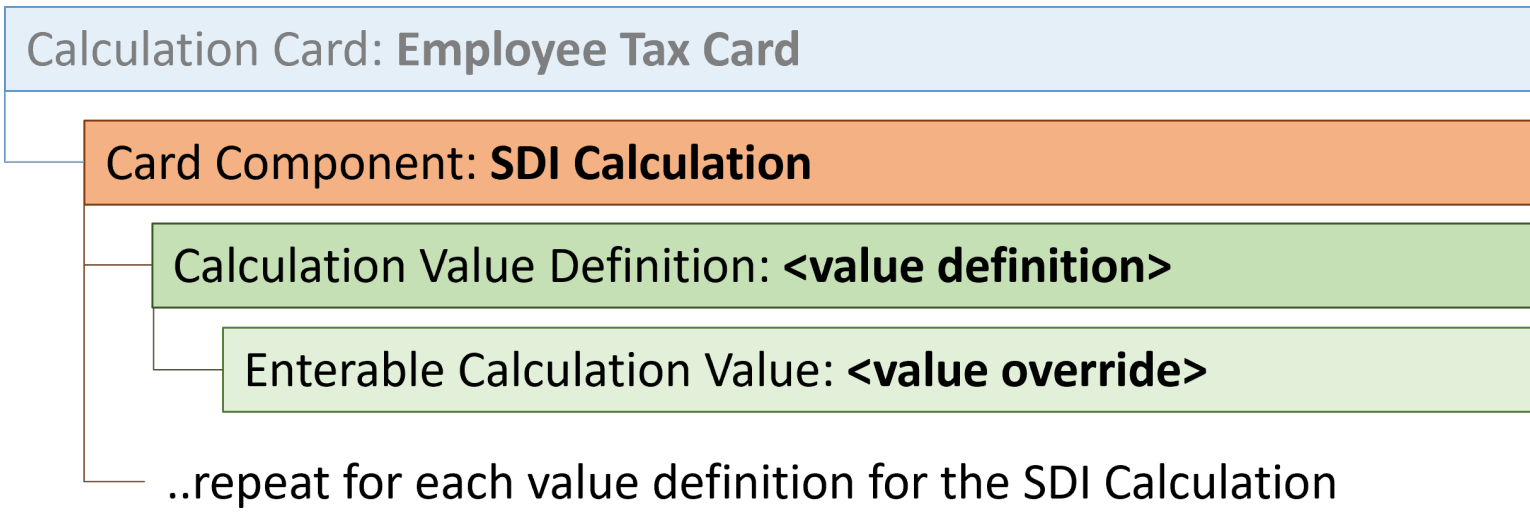
```
METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|ETC_234116_INF_VD1|MX LDG|ETC_234116_INF|2023/01/01|Employee
 INFONAVIT|Discount Type
MERGE|CalculationValueDefinition|VISION|ETC_234116_INF_VD2|MX LDG|ETC_234116_INF|2023/01/01|Employee
 INFONAVIT|Credit Number
MERGE|CalculationValueDefinition|VISION|ETC_234116_INF_VD3|MX LDG|ETC_234116_INF|2023/01/01|Employee
 INFONAVIT|Credit Start date
MERGE|CalculationValueDefinition|VISION|ETC_234116_INF_VD4|MX LDG|ETC_234116_INF|2023/01/01|Employee
 INFONAVIT|Discount Value
MERGE|CalculationValueDefinition|VISION|ETC_234116_INF_VD5|MX LDG|ETC_234116_INF|2023/01/01|Employee
 INFONAVIT|Mexico SUA INFONAVIT Transaction

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|Value1|ValueDefinitionName
MERGE|EnterableCalculationValue|VISION|ETC_234116_INF_VD1|MX LDG|ETC_234116_INF_VD1|2023/01/01|Fix quota in
 currency|Discount Type
MERGE|EnterableCalculationValue|VISION|ETC_234116_INF_VD2|MX LDG|ETC_234116_INF_VD2|2023/01/01|100123|Credit
 Number
MERGE|EnterableCalculationValue|VISION|ETC_234116_INF_VD3|MX LDG|ETC_234116_INF_VD3|2023/01/01|2023/01/01|
Credit Start date
MERGE|EnterableCalculationValue|VISION|ETC_234116_INF_VD4|MX LDG|ETC_234116_INF_VD4|2023/01/01|20|Discount
 Value
MERGE|EnterableCalculationValue|VISION|ETC_234116_INF_VD5|MX LDG|ETC_234116_INF_VD5|2023/01/01|15 -
 Beginning of housing credit|Mexico SUA INFONAVIT Transaction
```

Guidelines for Loading SDI Calculation for the Mexico Employee Tax Card

The SDI Calculation card component captures all information around the employee's base salary, which is used for social insurance calculations, including contributions to social security and INFONAVIT loan paybacks.

SDI Calculation Card Component

Use this card component to load exceptions in case you have to override the values stored at the organization card level.



Card Component Attributes for SDI Calculation

The SDI Calculation card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the SDI Calculation card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Tax Cardcalculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the SDI Calculation card component, typically the employee's start date. This must be on or after the EffectiveStartDate on the Employee Tax Card calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'SDI Calculation'.
ComponentSequence	N/A	Specify '1'.

SDI Calculation Value Definitions

The SDI Calculation card component uses value definitions to supply override values.

The rates or values displayed in the Employee Tax Card for SDI Calculation are the default values defined at the legislative level. They're provided on the Employee Tax Card for reference purposes.

Value Definition Name	Functional Description
SDI Variable Portion Override	Use the SDI Variable Portion Override to specify an employee's previous bi-monthly variable earnings. This override amount will be used for the calculation of SDI (integrated daily wage). This can also be used for any new hires/rehires where variable earnings form part of the employee's expected earnings.
Annual pay period unit	Employee's unit of annual pay period.
Number of biweekly periods in the year	Number of biweekly periods in the year, which triggers the employee's SDI calculation.

Value Definition Name	Functional Description
Number of days in the year	Employee's number of days in the year
Number of hours in the year	Employee's number of hours in the year
Number of months in the year	Employee's number of months in the year
Number of semi-monthly periods in the year	Employee's number of semi-monthly periods in the year
Number of ten-days periods in the year	Employee's number of ten-day periods in the year.
Number of weeks in the year	Employee's number of weeks in the year.

Calculation Value Definition Attributes for SDI Calculation

The Calculation Value Definition record type specifies the name of the value definition you are supplying an override value for.

The Calculation Value Definition record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the calculation value definition record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
SourceId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent SDI Calculation card component should be referenced by using the same key type used to identify the card component. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the parent SDI Calculation card component or the date the calculation value definition starts, if later.
DirCardCompDefName	N/A	The definition name of the parent SDI Calculation card component. Specify the same value as provided on the parent card component record.
ValueDefinitionName	N/A	The name of the value being overridden. The list of value definitions applicable to this card component are listed above.

These attributes are supplied against the CalculationValueDefinition file discriminator and must be supplied along with a CardComponent record for the parent SDI Calculation card component and a CalculationCard record for the owning Employee Tax Card.

Enterable Calculation Value Attributes for Employee INFONAVIT

The Enterable Calculation Value provides the override value for the value definition. It references the Calculation Value Definition record which defines the Value Definition being overridden.

The Enterable Calculation Value record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the enterable calculation value record. For new records supply the source key attributes. You can also identify calculation value definition records with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the override value.
ValueDefnId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName, ValueDefinitionName	Identify the parent Calculation Value Definition record using the same key type used to identify the calculation value definition. When using source keys, supply this attribute with the value supplied for the calculation value definition's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent record.
EffectiveStartDate	N/A	The effective start date of the parent calculation value definition record, or the update to the override value if supplying date-effective history.
EffectiveEndDate	N/A	The optional end date of the override value, or if you are providing date-effective history, the last day of the date-effective changes
Value1	N/A	The value for the value definition identified by the parent calculation value definition record. Unlike other calculation cards, if you supply a value definition but have no value for it supply '-999999999' to indicate a null value.

These attributes are supplied against the EnterableValueDefinition file discriminator. You must supply an EnterableValueDefinition record for each CalculationValueDefi.

Example of Creating SDI Calculation Card Component

This example creates the SDI Calculation component for the existing Employee Tax Card calculation card belonging to 3 employees with the assignment numbers E8MXSDI19Nov, E9MXSDI19Nov and E10MXSDI19Nov.

A value (150) is being entered for the SDI Variable Portion Override value definition that belongs to the SDI Calculation component. User keys are provided to reference the existing Employee Tax Card and Employee Taxation card component on the assumption they were auto generated and therefore the source key values are unknown.

The CalculationCard.dat file is used to upload the SDI Calculation component and provide the value for the SDI Variable Portion Override value definition with HCM Data Loader.

```
METADATA|CalculationCard|EffectiveStartDate|EffectiveEndDate|LegislativeDataGroupName|DirCardDefinitionName|
AssignmentNumber|CardSequence
MERGE|CalculationCard|2017/12/01|4712/12/31|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E8MXSDI19Nov|1
MERGE|CalculationCard|2020/01/01|4712/12/31|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E9MXSDI19Nov|1
MERGE|CalculationCard|2020/01/01|4712/12/31|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E10MXSDI19Nov|1

METADATA|CardComponent|EffectiveStartDate|EffectiveEndDate|LegislativeDataGroupName|DirCardDefinitionName|
AssignmentNumber|CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|2017/12/01|4712/12/31|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E8MXSDI19Nov|1|SDI
 Calculation|1
MERGE|CardComponent|2020/01/01|4712/12/31|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E9MXSDI19Nov|1|SDI
 Calculation|1
MERGE|CardComponent|2020/01/01|4712/12/31|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E10MXSDI19Nov|1|SDI
 Calculation|1

METADATA|CalculationValueDefinition|EffectiveStartDate|EffectiveEndDate|LegislativeDataGroupName|
DirCardDefinitionName|AssignmentNumber|CardSequence|DirCardCompDefName|ComponentSequence|ValueDefinitionName
MERGE|CalculationValueDefinition|2020/04/01|2020/04/30|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E8MXSDI19Nov|
1|SDI Calculation|1|SDI Variable Portion Override
MERGE|CalculationValueDefinition|2020/01/01|2020/02/29|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E9MXSDI19Nov|
1|SDI Calculation|1|SDI Variable Portion Override
MERGE|CalculationValueDefinition|2020/01/01|2020/03/31|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E10MXSDI19Nov|
1|SDI Calculation|1|SDI Variable Portion Override

METADATA|EnterableCalculationValue|EffectiveStartDate|EffectiveEndDate|LegislativeDataGroupName|
DirCardDefinitionName|AssignmentNumber|CardSequence|DirCardCompDefName|ComponentSequence|
ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|2020/04/01|2020/04/30|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E8MXSDI19Nov|1|
SDI Calculation|1|SDI Variable Portion Override|150
MERGE|EnterableCalculationValue|2020/01/01|2020/02/29|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E9MXSDI19Nov|1|
SDI Calculation|1|SDI Variable Portion Override|150
MERGE|EnterableCalculationValue|2020/01/01|2020/03/31|ZHRX_MX_RRF_LDG_01VK|Employee Tax Card|E10MXSDI19Nov|
1|SDI Calculation|1|SDI Variable Portion Override|150
```

Guidelines for Associating Employee Tax Cards for Mexico

The card association record associates the Employee Tax Card calculation card with the employee's tax reporting unit.

The associated tax reporting unit is defined in the card association. The card association detail allows the Employee Taxation card component to be associated with the employee's assignments.

Card Association Attributes for Employee Tax Card

The Card Association record type associates the Employee Tax Card calculation card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Employee Tax Card card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Tax Card calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Tax Card calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Employee Tax Card calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Employee Taxation

The card association detail associates the Employee Taxation card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Thecard component this association is for. Supply either the source system ID value or user key attributes to identify the Employee Taxation card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Example of Loading Employee Tax Card Associations

This example creates the tax reporting unit association for the existing Employee Tax Card calculation card belonging to employee assignment E95516.

User keys are provided to reference the existing Employee Tax Card and Employee Taxation card component on the assumption they were auto generated and therefore the source key values are unknown.

The CalculationCard.dat file is used to upload the Employee Tax Card card associations with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|MX LDG|Employee Tax Card|2023/01/01|E95516|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|MX LDG|Employee Tax Card|2023/01/01|E95516|1|Employee Taxation|1

METADATA|CardAssociation|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|TaxReportingUnitName
MERGE|CardAssociation|MX LDG|Employee Tax Card|2023/01/01|E95516|1|MX Tax Reporting Unit

METADATA|CardAssociationDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|
AssignmentNumber|CardSequence|TaxReportingUnitName|DirCardCompDefName|ComponentSequence|
AssociationAssignmentNumber
MERGE|CardAssociationDetail|MX LDG|Employee Tax Card|2023/01/01|E95516|1|MX Tax Reporting Unit|Employee
 Taxation|1|E95516
```

Examples of Creating Employee Tax Card for Mexico

These examples show how employee tax cards can be created.

Creating a Default Employee Tax Card

This example creates an Employee Tax Card calculation card with associations for employee assignment E1515. If your Employee Tax Card cards are not auto-generated it is recommended that you use source keys to create the card.

The CalculationCard.dat file is used to upload Employee Tax Card calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|ETC_1515|MX LDG|Employee Tax Card|2023/01/01|E1515

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|ETC_1515_ISR|MX LDG|ETC_1515|2023/01/01|Employee ISR Tax
MERGE|CardComponent|VISION|ETC_1515_SS|MX LDG|ETC_1515|2023/01/01|Employee Social Security
MERGE|CardComponent|VISION|ETC_1515_ET|MX LDG|ETC_1515|2023/01/01|Employee Taxation

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|ETC_1515_TRU|MX LDG|ETC_1515|2023/01/01|MX Tax Reporting Unit

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirRepCardId(SourceSystemId)|EffectiveStartDate|DirCardCompId(SourceSystemId)|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|ETC_1515_AD|MX LDG|ETC_1515_TRU|2023/01/01|ETC_1515_ET|E1515
```

Creating an Employee Tax Card with Overrides

This example creates an Employee Tax Card calculation card with associations for employee assignment E234116.

The Employee ISR Tax card component is updated, overriding this value:

Value Definition	Value
Pay Subsidy For Employment	No

The Employee Social Security card component is updated, overriding these values:

Value Definition	Value
Additional Quota Exempt	No
Additional Quota Percentage	1

If your Employee Tax Card cards are not auto-generated it is recommended that you use source keys to create the card.

The CalculationCard.dat file is used to upload Employee Tax Card calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|ETC_234116|MX LDG|Employee Tax Card|2023/01/01|E234116
```

```
METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|ETC_234116_ISR|MX LDG|ETC_234116|2023/01/01|Employee ISR Tax
MERGE|CardComponent|VISION|ETC_234116_ET|MX LDG|ETC_234116|2023/01/01|Employee Taxation
MERGE|CardComponent|VISION|ETC_234116_SS|MX LDG|ETC_234116|2023/01/01|Employee Social Security

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|ETC_234116_TRU|MX LDG|ETC_234116|2023/01/01|MX Tax Reporting Unit

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirRepCardId(SourceSystemId)|EffectiveStartDate|DirCardCompId(SourceSystemId)|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|ETC_234116_AD|MX LDG|ETC_234116_TRU|2023/01/01|ETC_234116_ET|E234116

METADATA|CalculationValueDefinition|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
SourceId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|ValueDefinitionName
MERGE|CalculationValueDefinition|VISION|ETC_234116_ISR_VD1|MX LDG|ETC_234116_ISR|2023/01/01|Employee ISR
 Tax|Pay Subsidy For Employment
MERGE|CalculationValueDefinition|VISION|ETC_234116_SS_VD1|MX LDG|ETC_234116_SS|2023/01/01|Employee Social
 Security|Additional Quota Exempt
MERGE|CalculationValueDefinition|VISION|ETC_234116_SS_VD2|MX LDG|ETC_234116_SS|2023/01/01|Employee Social
 Security|Additional Quota Percentage

METADATA|EnterableCalculationValue|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
ValueDefnId(SourceSystemId)|EffectiveStartDate|ValueDefinitionName|Value1
MERGE|EnterableCalculationValue|VISION|ETC_234116_ISR_VD1|MX LDG|ETC_234116_ISR_VD1|2023/01/01|Pay Subsidy
 For Employment|No
MERGE|EnterableCalculationValue|VISION|ETC_234116_SS_VD1|MX LDG|ETC_234116_SS_VD1|2023/01/01|Additional
 Quota Exempt|No
MERGE|EnterableCalculationValue|VISION|ETC_234116_SS_VD2|MX LDG|ETC_234116_SS_VD2|2023/01/01|Additional
 Quota Percentage|0.1
```

Initializing Balances

Balances to Initialize for Mexico

The Mexico Legislative Balances are what Oracle Payroll Cloud uses to perform accurate tax reporting and payroll tax calculations.

Mexico Balances

Oracle HCM Cloud requires several balances in order to perform crucial calculations pertaining to an employee for earnings and statutory deductions, like tax and social insurance, benefits, pensions, etc.

A list of balances you must consider follows. The dimension information refers to the following predefined balance dimensions:

- Year to Date: Relationship Tax Unit Year to Date
- Period to Date: Relationship Tax Unit Period to Date
- Month to Date: Relationship Tax Unit Month to Date
- Bi-Month to Date: Relationship Tax two Months to Date

Employee Tax Credits

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
ISR Subsidy For Employment Paid Regular	Yes	Yes	N/A	Yes

Employer Charges

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
Additional Quota ER	Yes	Yes	N/A	N/A
Benefits in Cash ER	Yes	Yes	N/A	N/A
Benefits in Species ER	Yes	Yes	N/A	N/A
Day Care Centers ER	Yes	Yes	N/A	N/A
Disability and Death ER	Yes	Yes	N/A	N/A
Disease and Maternity ER	Yes	Yes	N/A	N/A
Fixed Quota ER	Yes	Yes	N/A	N/A
INFONAVIT ER	Yes	Yes	N/A	N/A
Pensioners Medical Expenses ER	Yes	Yes	N/A	N/A
Retirement ER	Yes	Yes	N/A	N/A
Separation due to Age ER	Yes	Yes	N/A	N/A
Work Risk Incident ER	Yes	Yes	N/A	N/A

Information

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
Fiscal Arrears	Yes	Yes	N/A	N/A

Miscellaneous

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
Eligible Days for Fixed Earnings	Yes	Yes	N/A	Yes
Eligible Worked Days for Profit Sharing	Yes	Yes	N/A	N/A
Fixed Earnings	Yes	Yes	N/A	Yes

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
ISR Exempt	Yes	Yes	N/A	Yes
ISR Subject	Yes	Yes	N/A	Yes
ISR Subject for Severance	Yes	Yes	N/A	N/A
ISR Subject NonPeriodic	Yes	Yes	N/A	N/A
ISR Subsidy for Employment	Yes	Yes	N/A	Yes
ISR Subsidy for Employment Paid	Yes	Yes	N/A	Yes
ISR Withheld	Yes	Yes	N/A	N/A
ISR NonWorking Days	Yes	Yes	N/A	Yes
Reduce Regular Absence Days	Yes	Yes	N/A	Yes
Reduce Regular Absence Earnings	Yes	Yes	N/A	Yes
Reduce Regular Absence Hours	Yes	Yes	N/A	Yes
Reduce Regular Days	Yes	Yes	N/A	Yes
Reduce Regular Earnings	N/A	N/A	N/A	N/A
Reduce Regular Hours	N/A	N/A	N/A	N/A
Severance Variable Basis Eligible Compensation	Yes	Yes	N/A	N/A
Exempt for Severance Earnings	Yes	Yes	N/A	N/A
Subject for Severance Earnings	Yes	Yes	N/A	N/A
Severance Fixed Basis Eligible Compensation	Yes	Yes	N/A	N/A
Mexico Sickness statutory balance	Yes	Yes	N/A	N/A
Social Foresight Earnings	Yes	Yes	N/A	N/A
Mexico Social Foresight Earnings statutory balance	Yes	Yes	N/A	N/A
Average SDI	Yes	Yes	N/A	N/A
State tax exempt for bi-month tax rule	Yes	Yes	Yes	Yes
State tax full exempt	Yes	Yes	N/A	N/A
State tax full subject	Yes	Yes	N/A	N/A

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
Exempt for State Social Foresight Earnings	Yes	Yes	N/A	Yes
Subject for State Social Foresight Earnings	Yes	Yes	N/A	Yes
Variable Earnings	Yes	Yes	Yes	Yes

ORA_DAYS

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
Days Basis of Quotation 1	Yes	Yes	Yes	Yes
Days Basis of Quotation 2	Yes	Yes	Yes	Yes
Days Basis of Quotation 3	Yes	Yes	Yes	Yes

ORA_ER_TAXABLE_EARN

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
INFONAVIT Social Security	Yes	Yes	N/A	Yes
Additional Quota Social Security Max Basis	Yes	Yes	N/A	Yes
Benefits in Cash Social Security Max Basis	Yes	Yes	N/A	Yes
Benefits in Species Social Security Max Basis	Yes	Yes	N/A	Yes
Day Care Centers Social Security Max Basis	Yes	Yes	N/A	Yes
Disability and Death Social Security Max Basis	Yes	Yes	N/A	Yes
Disease and Maternity Social Security Max Basis	Yes	Yes	N/A	Yes
Fixed Quota Social Security Max Basis	Yes	Yes	N/A	Yes
Pensioners Medical Expenses Social Security Max Basis	Yes	Yes	N/A	Yes
Retirement Social Security Max Basis	Yes	Yes	N/A	Yes
Separation due to Age Social Security Max Basis	Yes	Yes	N/A	Yes

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
Work Risk Incident Social Security Max Basis	Yes	Yes	N/A	Yes

ORA_TAX_DEDUCTION_DETAIL

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
ISR Subsidy For Employment Regular	Yes	Yes	N/A	Yes

ORA_TOTAL_ABSENCES

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
Absences	Yes	Yes	Yes	Yes
Maternity	Yes	Yes	Yes	Yes
Other	Yes	Yes	Yes	Yes
Sickness	Yes	Yes	Yes	Yes
Vacation	Yes	Yes	Yes	Yes

ORA_TOTAL_SEVERANCE_ PAY

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
Severance Earnings	Yes	Yes	N/A	N/A
Severance Indemnity Earnings	Yes	Yes	N/A	N/A
Severance Seniority Premium Earnings	Yes	Yes	N/A	N/A

Social Insurance Deductions

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
Additional Quota EE	Yes	Yes	N/A	N/A
Benefits in Species EE	Yes	Yes	N/A	N/A
Day Care Centers EE	Yes	Yes	N/A	N/A
Fixed Quota EE	Yes	Yes	N/A	N/A
INFONAVIT EE	Yes	Yes	N/A	N/A

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
Pensioners Medical Expenses EE	Yes	Yes	N/A	N/A
Retirement EE	Yes	Yes	N/A	N/A
Separation due to Age EE	Yes	Yes	N/A	N/A
Work Risk Incident EE	Yes	Yes	N/A	N/A

Tax Deductions

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
ISR Severance	Yes	Yes	N/A	N/A
ISR NonPeriodic	Yes	Yes	N/A	N/A
ISR Regular	Yes	Yes	N/A	Yes

Tax Earnings

Balance	Year to Date	Period to Date	Month to Date	Bi-Month to Date
Gross Earnings	Yes	Yes	Yes	Yes

Profit Sharing

The Profit Sharing payment is paid to employees in May following the year that profit sharing was calculated for. For example, the 2022 fiscal year profit sharing will be paid in May 2023.

The previous three years Profit Sharing payments are used to calculate the payment cap. To calculate the 2022 fiscal year profit sharing, the employee's 2019, 2020 and 2021 profit sharing payments are used.

To ensure the accuracy of the Mexico profit sharing calculation, you will need to initialise or adjust these profit sharing balances.

Balance	Dimension
Mexico Profit Sharing statutory balance	Relationship NoCB Year to Date
Eligible Compensation for Profit Sharing	Relationship NoCB Year to Date
Eligible Worked Days for Profit Sharing	Relationship NoCB Year to Date

Other Balances

Depending on your requirements you should also consider initializing the following:

- Run balances, if your project has a go-live date that falls in the middle of a payroll period.

- Inception to date balances needed by your organization
- User defined balances.

Note: When initializing Inception to Date balances, the Payroll referenced in the batch must exist in the previous year and have payroll periods defined for that previous year (because the process will attempt to create a balance initialization in the previous year).

Related Topics

- Overview of Balance Initialization
- Steps to Initialize Balances

Balance Initialization Scenarios for Mexico

Depending on the go-live date, you need to initialize specific balances, to avoid inaccurate results in payroll calculations.

Go-live Mid-month

COMPONENT	SUB-COMPONENT	DESCRIPTION	BALANCES TO INITIALIZE	DIMENSIONS
ISR Tax for Regular	Subsidy for Employment	ISR Subsidy for Employment is adjusted automatically every pay period to ensure that the monthly cap is not exceeded.	• Earnings for ISR • ISR NonWorking Days ISR Subject • ISR Subsidy for Employment • ISR Subsidy for Employment Paid ISR Subsidy For Employment Paid Regular	Relationship Tax Unit Month to Date Relationship Tax Unit Period to Date
Monthly Tax Adjustment	N/A	The Monthly Tax Adjustment Payroll Run Type: • Prorates monthly rates based on the pay period it isused. • Computes the tax for the accumulated ISR subject earnings earned in the current period andall previous periods in the month. • Adjusts the ISR Withheldbalance • Withholds/reimburses the difference in the payroll run	• Fixed Earnings ISR Calculated • ISR NonWorking Days ISR Subject • ISR Subsidy for Employment • ISR Subsidy for Employment Paid ISR Tax to Charge • ISR Withheld	Relationship Tax Unit Month to Date Relationship Tax Unit Period to Date

COMPONENT	SUB-COMPONENT	DESCRIPTION	BALANCES TO INITIALIZE	DIMENSIONS
ISR Tax for Non-Periodic ISR Tax for Severance	Monthly Ordinary Salary	The ISR Tax calculation methods for Severance and Non-periodic earnings are dependent on the employee's monthly salary.	• Fixed Earnings • Eligible Days for Fixed Earnings	Relationship Tax Unit Month to Date Relationship Tax Unit Period to Date

Go-live Mid-year

COMPONENT	SUB-COMPONENT	DESCRIPTION	BALANCES TO INITIALIZE	DIMENSIONS
Social Foresight Earnings taxability rule	N/A	The Social Foresight Taxability rule is computed based on the total annual employee income and the Social Foresight earnings.	• Gross Earnings • Social Foresight Earnings	Relationship Tax Unit Year to Date
Profit sharing	N/A	The current Fiscal Year (FY) profit sharing is processed and paid every May of the next FY. (For example, FY 2022 profit sharing is paid in May 2023). In order to determine the average of the employee's profit sharing participation received in the last three years, the Mexico Profit Sharing statutory balance needs to be initialized for each of the employee's previous 3 payments.	• Mexico Profit Sharing statutory balance • Eligible Compensation for Profit Sharing • Eligible Worked Days for Profit Sharing	Relationship NoCB Year to Date

Go-live Beginning of Bi-month

COMPONENT	SUB-COMPONENT	DESCRIPTION	BALANCES TO INITIALIZE	DIMENSIONS
SDI Variable Portion	N/A	The current bi-month SDI variable portion is derived from the variable compensation earned in the previous bi-month on a daily basis.	Variable Earnings Days Basis of Quotation 1	Relationship Tax 2 Months to Date

Go-live Middle of Bi-month

COMPONENT	SUB-COMPONENT	DESCRIPTION	BALANCES TO INITIALIZE	DIMENSIONS
SDI Variable Portion	N/A	In addition to the previous bi-month balance that is needed to compute the current bi-month variable	• Gross Earnings • Social Foresight Earnings	• Relationship Tax 2 Months to Date

COMPONENT	SUB-COMPONENT	DESCRIPTION	BALANCES TO INITIALIZE	DIMENSIONS
		portion, it will be necessary to initialize the balance with the pay-period values of the current bi-month to be able to compute next bi-month SDI value.		• Relationship Tax Unit Month to Date • Relationship Tax Unit Period to Date
Employer State Tax Oaxaca State	Taxability rules	Oaxaca (OAX) state has an exemption tax rule 'The exemption will be up to one Employee Minimum Wage elevated to the current Bi-Month', applicable for some secondary classifications.	• Glasses Assistance Educational Assistance • Scholarship for Children of Workers Scholarship for Workers • Healthcare Reimbursement Social Foresight Earnings	• Relationship Tax Unit, Area1 Period to Date • Relationship Tax Unit, Area1 Run • Relationship Tax Unit, Area1 Year to Date

Example of Initializing Employee Balances for Mexico

In this example, you create balance initialization records to initialize an employee's Fixed Earnings and Profit Sharing related balances.

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|LegislativeDataGroupName|BatchName|UploadDate
MERGE|InitializeBalanceBatchHeader|MX LDG|MX Balance Initialization|2023/02/28
```

Load the Initialize Balance Batch Lines

In the following example a number of balances are being initialized for the same employee. The employee is identified by their payroll relationship number, employment terms number and assignment number.

Attribute	Value
PayrollRelationshipNumber	192051
TermNumber	ET192051
AssignmentNumber	E192051

The employee's payroll, legal employer and tax reporting unit must also be specified on every initialize balance line

Attribute	Value
PayrollName	MX Monthly
LegalEmployerName	MX Legal Employer

Attribute	Value
TaxUnitName	MX Tax Reporting Unit

Load these details to initialize the Fixed Earnings Balance to 12000:

Attribute	Value
LineSequence	1
BalanceName	Fixed Earnings
DimensionName	Relationship Tax Unit Month to Date
Value	12000

Load these details to initialize the Eligible Days for the Fixed Earnings Balance to 23:

Attribute	Value
LineSequence	2
BalanceName	Eligible Days for Fixed Earnings
DimensionName	Relationship Tax Unit Month to Date
Value	23

Load these details to initialize the Eligible Worked Days for Profit Sharing balance to 23:

Attribute	Value
LineSequence	3
BalanceName	Eligible Worked Days for Profit Sharing
DimensionName	Relationship Tax Unit Month to Date
Value	23

Load these details to initialize the Eligible Compensation for Profit Sharing balance to 3000:

Attribute	Value
LineSequence	4
BalanceName	Eligible Compensation for Profit Sharing
DimensionName	Relationship Tax Unit Month to Date
Value	3000

Use the InitializeBalanceBatchLine.dat file to load the balance initialization lines defined above:

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|UploadDate|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|LegalEmployerName|PayrollName|TaxUnitName|BalanceName|
DimensionName|Value
MERGE|InitializeBalanceBatchLine|MX LDG|MX Balance Initialization|2023/02/28|1|192051|ET192051|E192051|MX
 Legal Employer|MX Monthly|MX Tax Reporting Unit|Fixed Earnings|Relationship Tax Unit Month to Date|12000
MERGE|InitializeBalanceBatchLine|MX LDG|MX Balance Initialization|2023/02/28|2|192051|ET192051|E192051|MX
 Legal Employer|MX Monthly|MX Tax Reporting Unit|Eligible Days for Fixed Earnings|Relationship Tax Unit
 Month to Date|23
MERGE|InitializeBalanceBatchLine|MX LDG|MX Balance Initialization|2023/02/28|3|192051|ET192051|E192051|MX
 Legal Employer|MX Monthly|MX Tax Reporting Unit|Eligible Worked Days for Profit Sharing|Relationship Tax
 Unit Month to Date|23
MERGE|InitializeBalanceBatchLine|MX LDG|MX Balance Initialization|2023/02/28|4|192051|ET192051|E192051|MX
 Legal Employer|MX Monthly|MX Tax Reporting Unit|Eligible Compensation for Profit Sharing|Relationship Tax
 Unit Month to Date|3000
```

Note: When initializing specific balances, you may also need to initialize any related balances. This is simply an example to illustrate the process.

Related Topics

- Overview of Balance Initialization
- Steps to Initialize Balances
- Overview of Balance Initialization for the US

28 Loading Payroll Localization Data for Bahrain

Employee Social Insurance

Overview of Employee Social Insurance Details for Bahrain

The Employee Social Insurance Details card stores all information required to accurately compute social insurance contributions, such as employee details, annuity information and reference salary.

Considerations and Prerequisites

When the product license is set to Payroll or Payroll Interface, Employee Social Insurance Details cards are automatically created when a new employee is entered using the New Hire task with a set of values specified at the person level, such as citizenship and social insurance information. The Social Insurance Details card component is also auto-generated. You only populate the component details if you need to supply override values.

There may be cases where this information needs to be loaded in bulk:

During Data Migration:

Employee Social Insurance information must be uploaded into Oracle Fusion Payroll, to ensure that contributions are calculated correctly. If HCM Data Loader is used to migrate employee records, a default Employee Social Insurance card is automatically created. In most cases, the default won't reflect the employee's actual Social Insurance information, and therefore the card must be updated.

Ongoing Bulk Updates:

- When you bulk load New Hire information, a default Employee Social Insurance card may be automatically generated (if the new hire records are created through HCM Data Loader or the interface with Taleo). In this case, you need to update the default card with the correct social insurance information.

It is recommended to have a good understanding of the Employee Social Insurance card and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS (Bahrain): Payroll Implementation and Functional Considerations (Document ID 2813681).

Employee Social Insurance Calculation Card Record Types

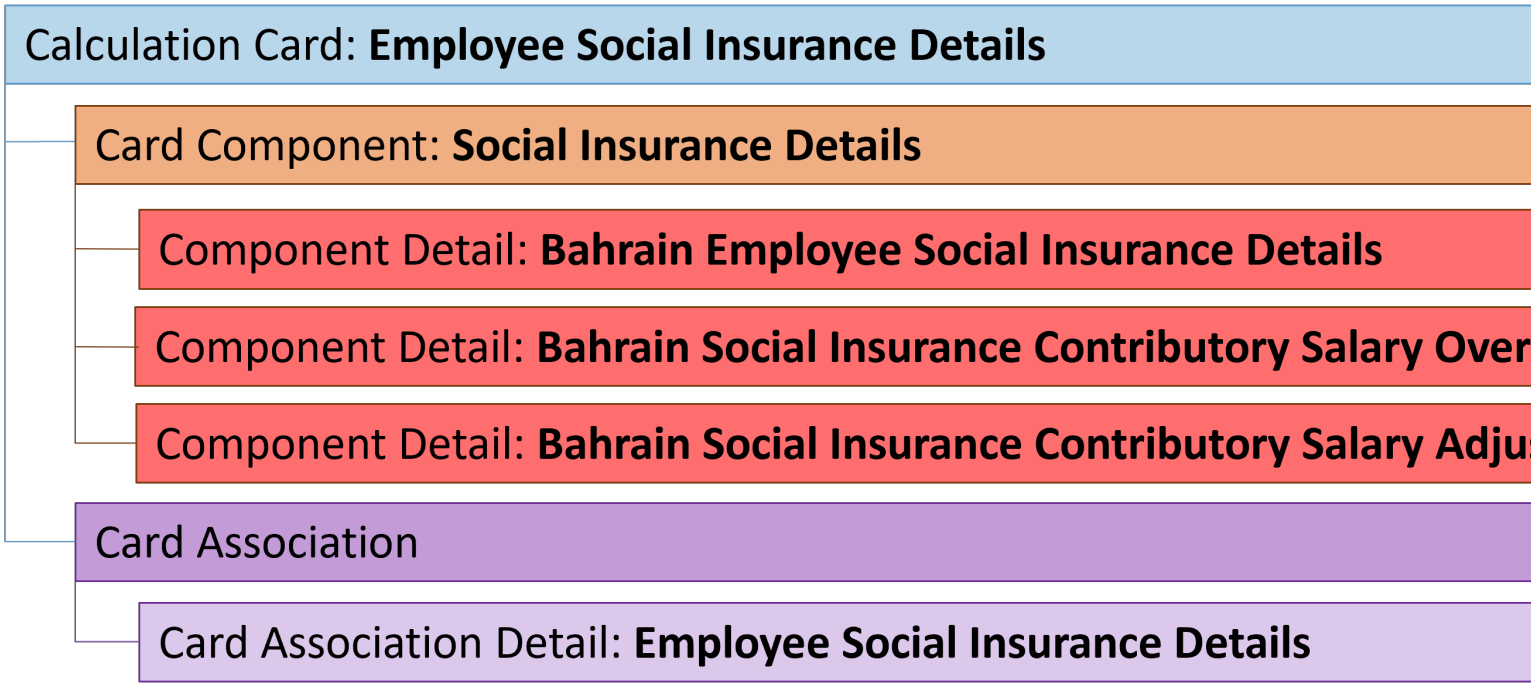
The Employee Social Insurance card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Employee Social Insurance utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Employee Social Insurance Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Employee Social Insurance are described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Updating Employee Social Insurance Details for Bahrain
- Guidelines for Associating Employee Social Insurance Details Cards for Bahrain
- Examples of Updating Employee Social Insurances Details for Bahrain
- Guidelines for Loading Flexfield Data

Guidelines for Updating Employee Social Insurance Details for Bahrain

You must have one Calculation Card record for every Bahrain employee you are maintaining Social Insurance Detail.

Even if you are updating an existing Employee Social Insurance Details card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

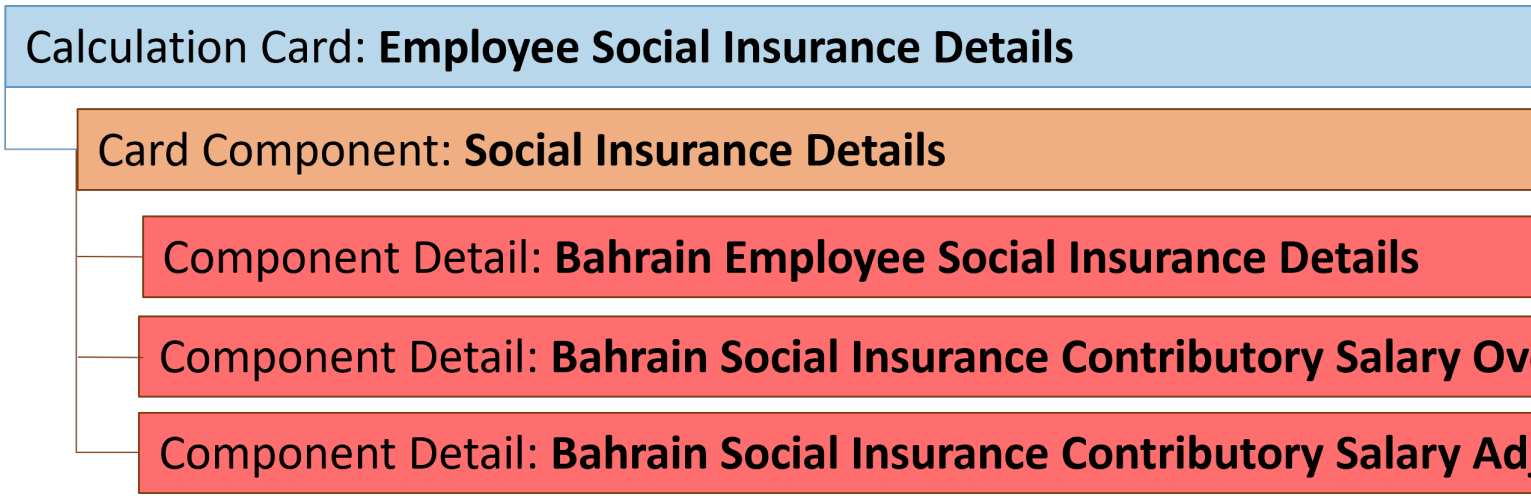
Employee Social Insurance Details Calculation Card Attributes

The Employee Social Insurance Details calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Employee Social Insurance Details calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Employee Social Insurance Details'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Social Insurance Details Card Component

The Social Insurance Details card component captures social insurance related information. It is created for you if the Employee Social Insurance Details card is autogenerated. If you need to update the defaulted social insurance information, provide a Social Insurance Details card component and related Component Detail.



Card Component Attributes for Employee Social Insurance Details Information

The Employee Social Insurance Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee Social Insurance Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Social Insurance Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Social Insurance Details card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Social Insurance Details calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Social Insurance Details'.
ComponentSequence	N/A	Specify '1'

Component Detail Attributes for Employee Social Insurance Details

Social Insurance Details card component uses three flexfield contexts. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Bahrain Employee Social Insurance Details (ORA_HRX_BH_SI_PREL)
- Bahrain Social Insurance Contributory Salary Override (ORA_HRX_BH_SI_CONT_SAL_OVR_PREL)
- Bahrain Social Insurance Contributory Salary Adjustment (ORA_HRX_BH_SI_CONT_SAL_ADJ_PREL)

Only the Bahrain Social Insurance Details context is required. The others are optional.

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Social Insurance Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		provided for the Social Insurance Details card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'ORA_HRX_BH_SI_PREL'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Overview of Employee Social Insurance Details for Bahrain
- Examples of Updating Employee Social Insurances Details for Bahrain

Examples of Updating Employee Social Insurances Details for Bahrain

These examples update or maintain an auto-generated Employee Social Insurance Details card.

User keys are used to identify the card and card component as the source keys may not be known for auto-generated cards. Use the CalculationCard.dat file to update Employee Social Insurance Details card component information with HCM Data Loader.

Example of Updating the Auto-Generated Employee Social Insurance Details Card Component

This example updates the existing Social Insurance Details card component for an auto-generated Employee Social Insurance Details card.

The Bahrain Social Insurance Details flexfield context is updated, setting these values:

Flexfield Segment	Value
Citizenship	Bahrain
Registered for Social Insurance	Yes
Exempt from SI Contribution Difference	Yes

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|BH LDG|Employee Social Insurance Details|2022/01/01|E3714|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|BH LDG|Employee Social Insurance Details|2022/01/01|E3714|1|Social Insurance Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
oraHrxBhCitizenship(Deduction Developer DF=ORA_HRX_BH_SI_PREL)|oraHrxBhRegisteredForSi(Deduction Developer
 DF=ORA_HRX_BH_SI_PREL)|oraHrxBhCtzenContryToPay(Deduction Developer DF=ORA_HRX_BH_SI_PREL)
MERGE|ComponentDetail|BH LDG|Employee Social Insurance Details|2022/01/01|E3714|1|Social Insurance Details|
1|ORA_HRX_BH_SI_PREL|ORA_HRX_BH_SI_PREL|BH|Y|Y
```

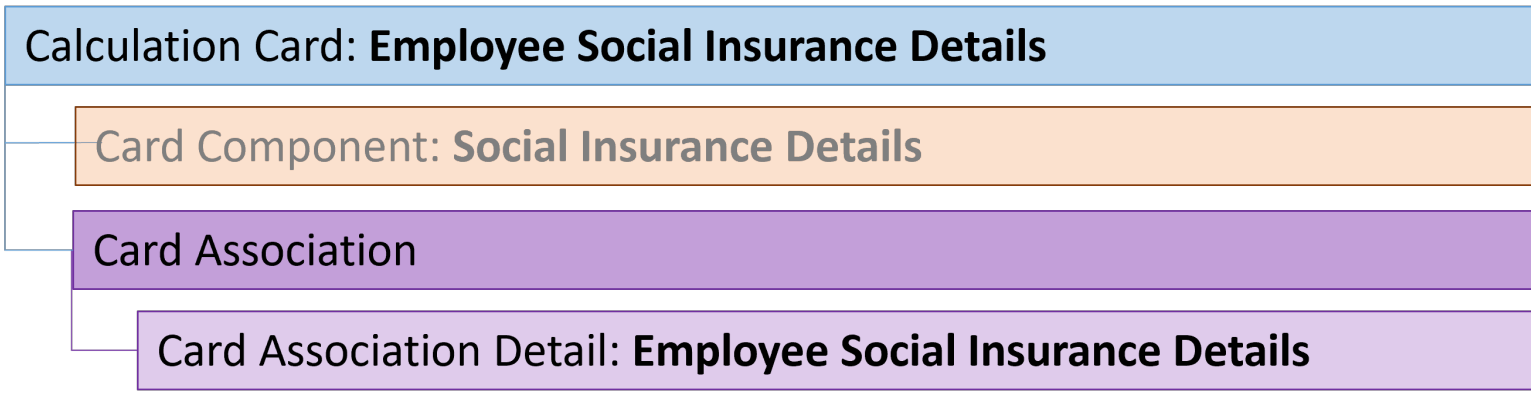
Example of Creating Bahrain Social Insurance Contributory Salary Adjustment

This example populates the Bahrain Social Insurance Contributory Salary Adjustment flexfield context for an auto-generated Employee Social Insurance Details card. The Adjustment Amount is set to 5000.00.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|BH LDG|Employee Social Insurance Details|2022/01/01|E3714|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|BH LDG|Employee Social Insurance Details|2022/01/01|E3714|1|Social Insurance Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
oraHrxBhContribSalAdjsmnt(Deduction Developer DF=ORA_HRX_BH_SI_CONT_SAL_ADJ_PREL)
MERGE|ComponentDetail|BH LDG|Employee Social Insurance Details|2022/01/01|E3714|1|Social Insurance Details|
1|ORA_HRX_BH_SI_CONT_SAL_ADJ_PREL|ORA_HRX_BH_SI_CONT_SAL_ADJ_PREL|5000.00
```

Guidelines for Associating Employee Social Insurance Details Cards for Bahrain

The card association record associates the Employee Social Insurance Details calculation card with the employee's tax reporting unit.



The associated tax reporting unit is defined in the card association. The card association detail allows the Social Insurance Details card component to be associated with the employee's assignments.

Card Association Attributes for Employee Social Insurance Details

The Card Association record type associates the Employee Social Insurance Details card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Employee Social Insurance Details card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Social Insurance Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Social Insurance Details calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Employee Social Insurance Details calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Employee Social Insurance Details

The card association detail associates the Social Insurance Details card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The card component this association is for. Supply either the source system ID value or user key attributes to identify the Social Insurance Details card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of Employee Social Insurance Details for Bahrain

Example of Loading Employee Social Insurance Details Card Associations for Bahrain

This example creates the tax reporting unit association for the existing Employee Social Insurance Details card belonging to employee assignment E3714.

User keys are provided to reference the existing Employee Social Insurance Details card and Social Insurance Details card component on the assumption they were auto generated and therefore the source key values are unknown.

The CalculationCard.dat file is used to upload the Employee Social Insurance Details card associations with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|BH LDG|Employee Social Insurance Details|2022/01/01|E3714|1
```

```
METADATA|CardAssociation|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|TaxReportingUnitName
MERGE|CardAssociation|BH LDG|Employee Social Insurance Details|2022/01/01|E3714|1|BH TRU

METADATA|CardAssociationDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|
AssignmentNumber|CardSequence|TaxReportingUnitName|DirCardCompDefName|ComponentSequence|
AssociationAssignmentNumber
MERGE|CardAssociationDetail|BH LDG|Employee Social Insurance Details|2022/01/01|E3714|1|BH TRU|Social
 Insurance Details|1|E3714
```

When associating an auto-generated card you don't need to include the CardComponent unless also making changes to its values.

Example of Creating Employee Social Insurance Details for Bahrain

This example creates an Employee Social Insurance Details card with associations for employee assignment E3714.

If your Employee Social Insurance Details cards are not auto-generated it is recommended that you use source keys to create the card.

The Bahrain Social Insurance Details flexfield context is populated with these values

Flexfield Segment	Value
Citizenship	Bahrain
Registered for Social Insurance	Yes
Exempt from SI Contribution Difference	Yes

The CalculationCard.dat file is used to upload Employee Social Insurance Details calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|3714_SI|BH LDG|Employee Social Insurance Details|2022/01/01|E3714

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|3714_SI_DET|BH LDG|3714_SI|2022/01/01|Social Insurance Details

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|oraHrxBhCitizenship(Deduction Developer DF=ORA_HRX_BH_SI_PREL)|
oraHrxBhRegisteredForSi(Deduction Developer DF=ORA_HRX_BH_SI_PREL)|oraHrxBhCtzenContryToPay(Deduction
 Developer DF=ORA_HRX_BH_SI_PREL)
MERGE|ComponentDetail|VISION|3714_SI_PREL|BH LDG|3714_SI_DET|2022/01/01|Social Insurance Details|
ORA_HRX_BH_SI_PREL|ORA_HRX_BH_SI_PREL|BH|Y|Y

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|3714_SI_TRU|BH LDG|3714_SI|2022/01/01|BH TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirRepCardId(SourceSystemId)|
DirCardCompId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
```

```
MERGE|CardAssociationDetail|VISION|3714_SI_TRU|3714_SI_TRU|3714_SI_DET|2022/01/01|E3714
```

Employee Gratuity Details

Overview of Employee Gratuity Details for Bahrain

The Employee Gratuity Details card stores all information required to calculate and process the gratuity payment, such as employee gratuity details (override amount and gratuity payment date).

The Employee Gratuity Details card is created automatically upon employee termination and the gratuity amount calculated. When the gratuity card is created, the Gratuity Details component, Bahrain Employee Gratuity Details component details and card association are also created.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, Employee Gratuity Details cards are automatically created upon employee termination. There may be cases where this information needs to be loaded in bulk:

During Data Migration:

Employee Gratuity Details information must exist into Oracle Fusion Payroll to ensure that values are calculated correctly. If HCM Data Loader is used to migrated employee termination records, a default Employee Gratuity Details card is automatically created. For historical reasons you may wish to upload gratuity calculation cards to records the Last Gratuity Payment Date for a previous employment.

Ongoing Bulk Updates:

- When you bulk load employee terminations a default Employee Gratuity Details calculation card may be generated automatically (if the terminations records are created using HCM Data Loader). In this case, you need to update the default card with the correct gratuity information and, if needed, create a component detail record to store a gratuity override amount

It is recommended to have a good understanding of the Employee Gratuity Details card and the information it contains prior to attempting mass upload as it has a direct impact on calculations and reporting. For further information, see Oracle Fusion HRMS (Bahrain): Payroll Implementation and Functional Considerations (Document ID 2813681.1).

Employee Gratuity Details Calculation Card Record Types

The Employee Gratuity Details card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

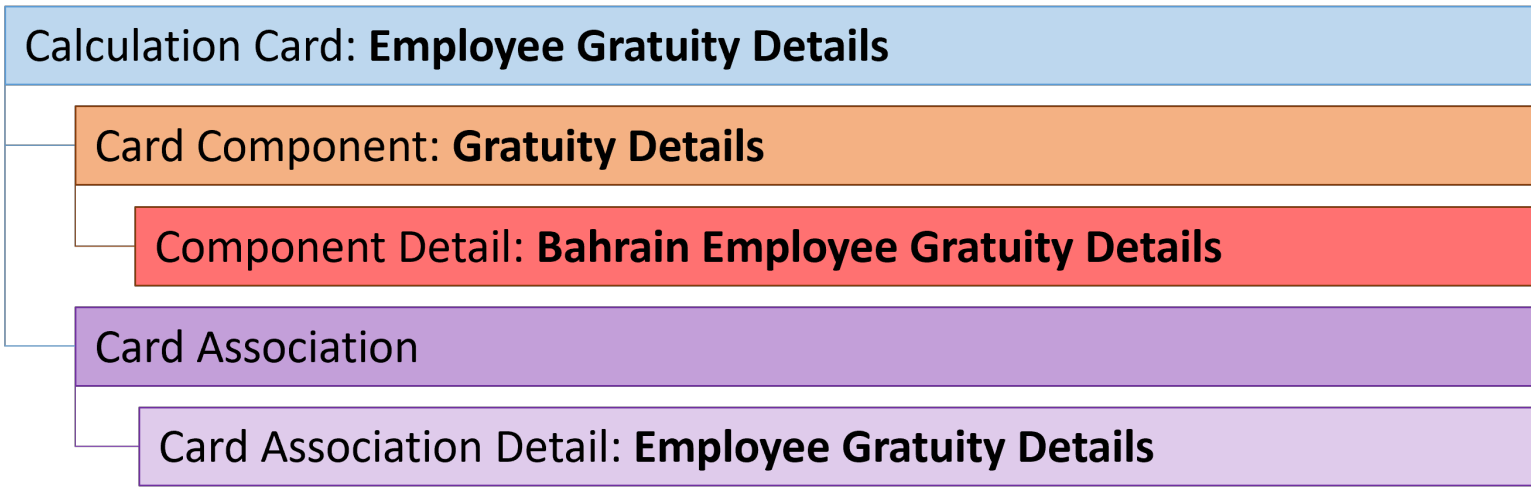
The Employee Gratuity Details utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this	CardComponent

Component	Functional Description	File Discriminator
	calculation card and the child records that are required for each card component.	
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Employee Gratuity Details Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Employee Gratuity Details are described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Employee Gratuity Details for Bahrain
- Guidelines for Associating Employee Gratuity Details Cards for Bahrain
- Example of Updating Employee Gratuity Details for Bahrain
- Guidelines for Loading Flexfield Data

Guidelines for Loading Employee Gratuity Details for Bahrain

You must have one Calculation Card record for every terminated Bahrain employee you are maintaining Gratuity Details for.

Even if you are updating an existing Employee Gratuity Details card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

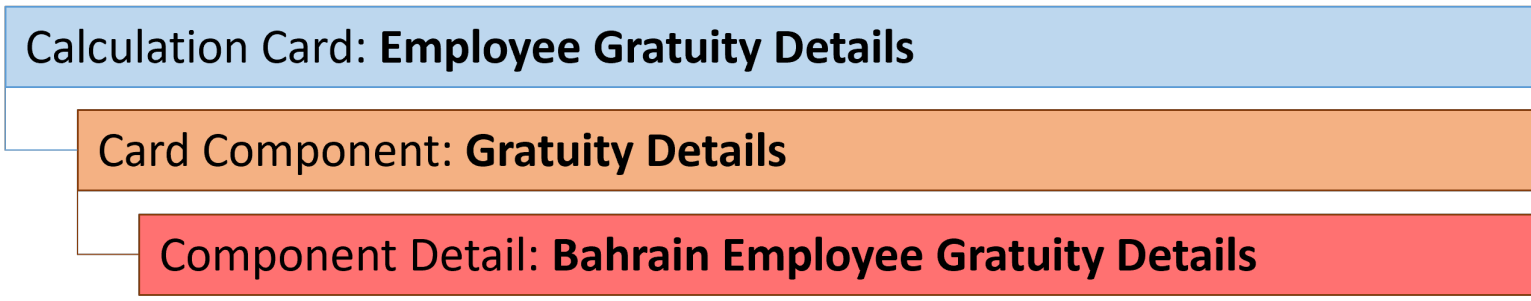
Employee Gratuity Details Calculation Card Attributes

The Employee Gratuity Details calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Employee Gratuity Details calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Employee Gratuity Details'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's termination date.
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Gratuity Details Card Component

The Gratuity Details card component captures social insurance related information. It is created for you if the Employee Gratuity Details card is autogenerated. If you need to update the defaulted gratuity information, provide a Gratuity Details card component and related Component Detail for each flexfield context relevant to the terminated employee.



Card Component Attributes for Gratuity Details

The Gratuity Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Gratuity Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Gratuity Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Gratuity Details card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Gratuity Details calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Gratuity Details'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.

Component Detail Attributes for Gratuity Details

Gratuity Details uses one flexfield context. Use the Component Detail record type to load the flexfield segment values. In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Bahrain Employee Gratuity Details (ORA_HRX_BH_GRATUITY_PREL)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Gratuity Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Gratuity Details card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as ORA_HRX_BH_GRATUITY_PREL. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Employee Gratuity Details for Bahrain
- Guidelines for Associating Employee Gratuity Details Cards for Bahrain
- Guidelines for Loading Flexfield Data

Example of Updating Employee Gratuity Details for Bahrain

This example updates an auto-generated Employee Gratuity Details card component of the Employee Gratuity Details card.

The Bahrain Employee Gratuity Details flexfield context is populated with these values for employee assignment E191668:

Flexfield Segment	Value
Override Amount	10050.00
Latest Gratuity Payment Date	31 Dec 2022

The CalculationCard.dat file is used to update Employee Gratuity Details calculation cards with HCM Data Loader.

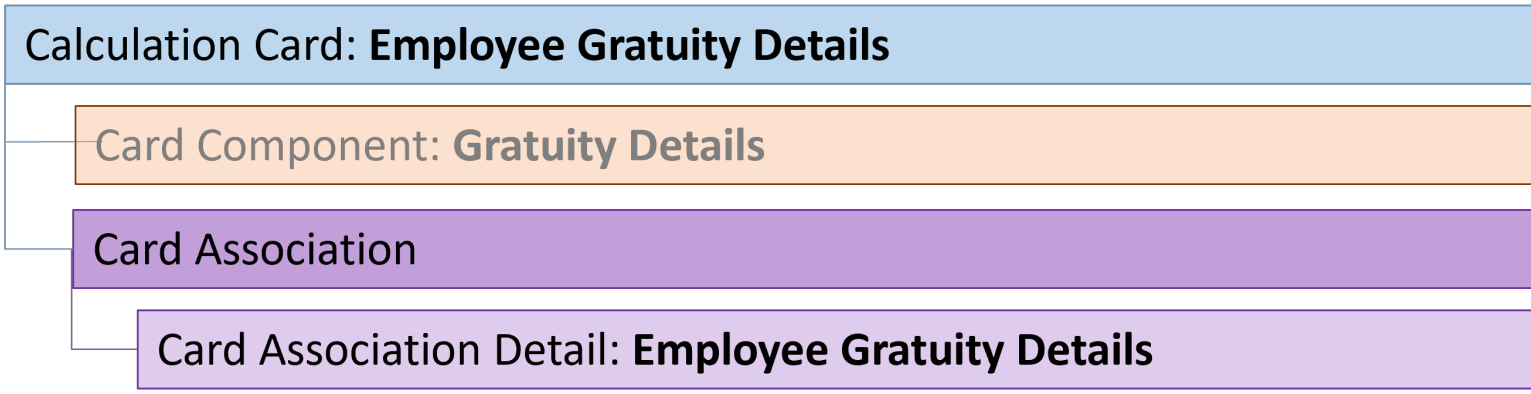
```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber
MERGE|CalculationCard|BH LDG|Employee Gratuity Details|1|2022/12/15|E3714

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|BH LDG|Employee Gratuity Details|1|2022/12/15|E3714|Gratuity Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
oraHrxBhOverrideAmount(Deduction Developer DF=ORA_HRX_BH_GRATUITY_PREL)|oraHrxBhLstGratPymtDate(Deduction
 Developer DF=ORA_HRX_BH_GRATUITY_PREL)
MERGE|ComponentDetail|BH LDG|Employee Gratuity Details|1|2022/12/15|E3714|Gratuity Details|1|
ORA_HRX_BH_GRATUITY_PREL|ORA_HRX_BH_GRATUITY_PREL|10050.00|2022/12/31
```

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Employee Gratuity Details for Bahrain
- Guidelines for Loading Flexfield Data

Guidelines for Associating Employee Gratuity Details Cards for Bahrain

The card association record associates the Employee Gratuity Details calculation card with the employee's tax reporting unit.



The associated tax reporting unit is defined in the card association. The card association detail allows the **Gratuity Details** card component to be associated with the employee's assignments.

Card Association Attributes for Employee Gratuity Details

The Card Association record type associates the **Employee Gratuity Details** card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Employee Gratuity Details card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Gratuity Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Gratuity Details calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Employee Gratuity Details calculation card. If source keys are used to identify the

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Gratuity Details

The card association detail associates the Gratuity Details card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Thecard component this association is for. Supply either the source system ID value or user key attributes to identify the Gratuity Details card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of Employee Gratuity Details for Bahrain
- Guidelines for Loading Employee Gratuity Details for Bahrain

Example of Creating Employee Gratuity Details for Bahrain

This example creates the Employee Gratuity Details calculation card with associations for employee assignment E3714 on the 1st January 2022, for a payment date of 31st December 2022.

If your Employee Gratuity Details cards are not auto-generated it is recommended that you use source keys to create the card.

The Bahrain Employee Gratuity Details flexfield context is populated with these values

Flexfield Segment	Value
Override Amount	20500.00
Latest Gratuity Payment Date	31 August 2022

The CalculationCard.dat file is used to upload Employee Gratuity Details calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|E3714_EGD|BH LDG|Employee Gratuity Details|2022/08/01|E3714

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|E3714_EGD_CMP|BH LDG|E3714_EGD|2022/08/01|Gratuity Details

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|oraHrxBhOverrideAmount(Deduction Developer DF=ORA_HRX_BH_GRATUITY_PREL)|
oraHrxBhLstGratPymtDate(Deduction Developer DF=ORA_HRX_BH_GRATUITY_PREL)
MERGE|ComponentDetail|VISION|E3714_EGD|BH LDG|E3714_EGD_CMP|2022/08/01|Gratuity Details|
ORA_HRX_BH_GRATUITY_PREL|ORA_HRX_BH_GRATUITY_PREL|20500.00|2022/08/31

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|E3714_EGD_ASS|BH LDG|E3714_EGD|2022/08/01| BH TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirRepCardId(SourceSystemId)|EffectiveStartDate|DirCardCompId(SourceSystemId)|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|E3714_EGD_ASD|BH LDG|E3714_EGD_ASS|2022/08/01|E3714_EGD_CMP|E3714
```

Court Orders

Overview of Court Orders for Bahrain

The Court Orders card stores information for court order deductions, such as amount and payee.

Considerations and Prerequisites

Use the Elements task to create Court Order elements. When you create court order elements, a card component definition is generated with the same name as the element. The generated card component definition is then available to use on the predefined Court Orders calculation card.

Unlike Employee Social Insurance or Pension Fund Details or End of Service Details cards, the Court Order card is not generated automatically. You only need to create this card for those employees who are subject to court order deductions.

There may be cases when there is a need to bulk load Court Orders cards:

During Data Migration:

Employee's Court Orders must bemigrated to the Oracle Payroll Cloud to ensure that appropriate deductions are consider. Using HCM Data Loader you must create the card and all necessary card components and component details.

Ongoing Bulk Updates:

- New hires may have Court Orders that need to be uploaded in bulk.

It is recommended to have a good understanding of the Court Orders card and the information it contains prior to attempting mass upload. For further information, see Oracle Fusion HRMS (Bahrain): Payroll Implementation and Functional Considerations (Doc ID 2813681).

Before bulk loading Court Order information:

- Create eligibility for the Court Order element.
- Create third-party payee and third-party payment methods for the people and organizations the payment of the court order is made to.
- Update the PAY_INVLN_DEDN_ORDER_AMOUNT_PAYEE lookup type with the court order issuing authorities.

Court Order Calculation Card Record Types

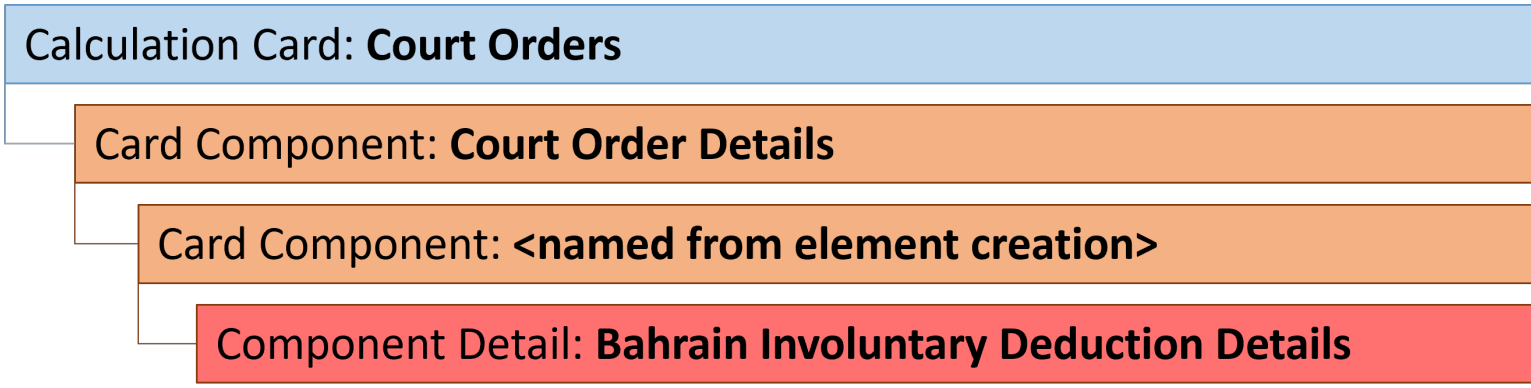
The Court Orders card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Court Orders card utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card comonoent.	ComponentDetail

Court Orders Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Court Orders are described in this diagram:

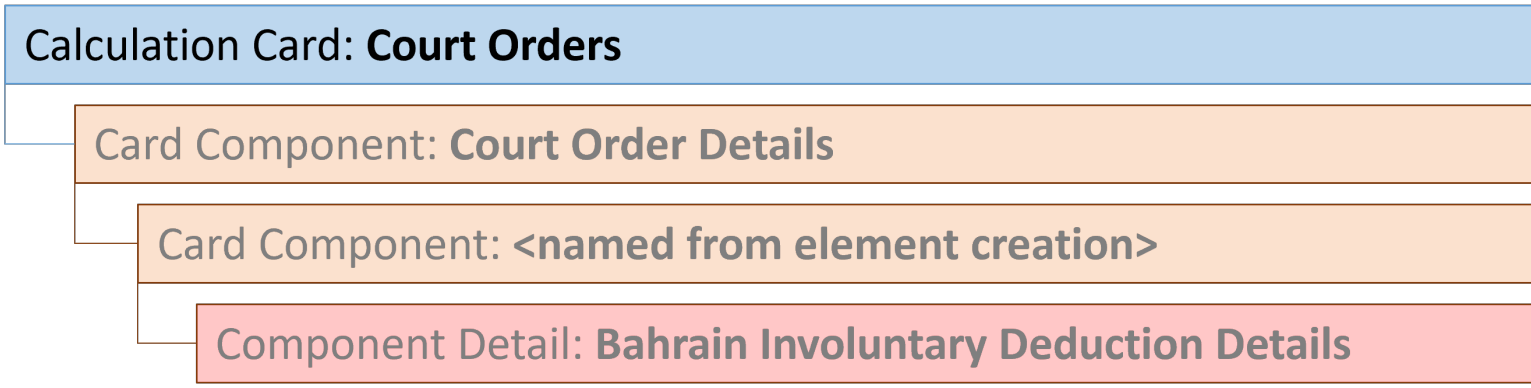


Related Topics

Guidelines for Loading Court Orders for Bahrain

You should always include the Court Orders calculation card record, even if you're updating an existing Court Orders card.

Even if the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.



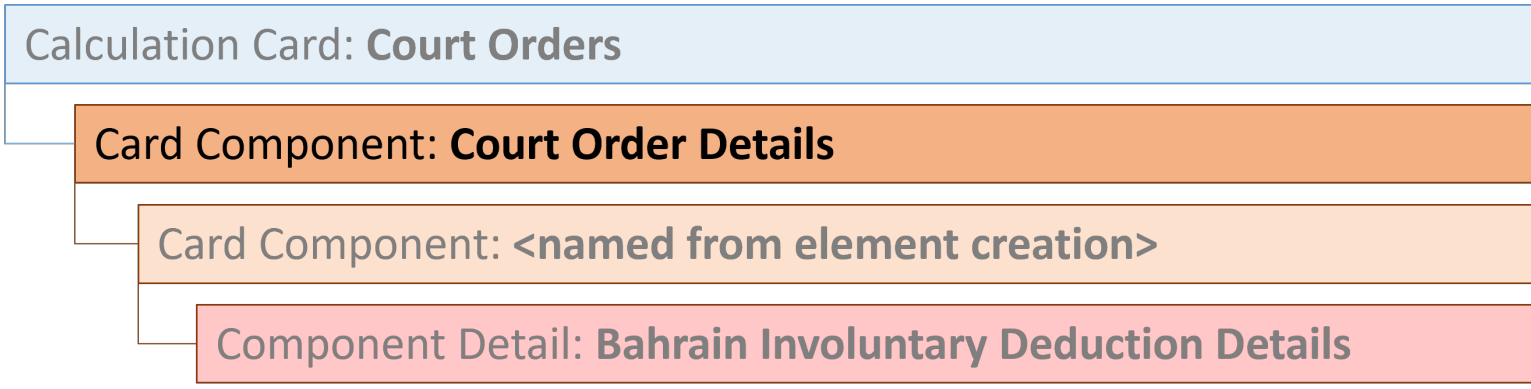
The Court Orders calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Court Orders calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Court Orders'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. **Note:** Not required when source keys are used.

Court Order Details Card Component

The Court Order Details card component is a predefined card component which acts as a parent to the element specific card component.



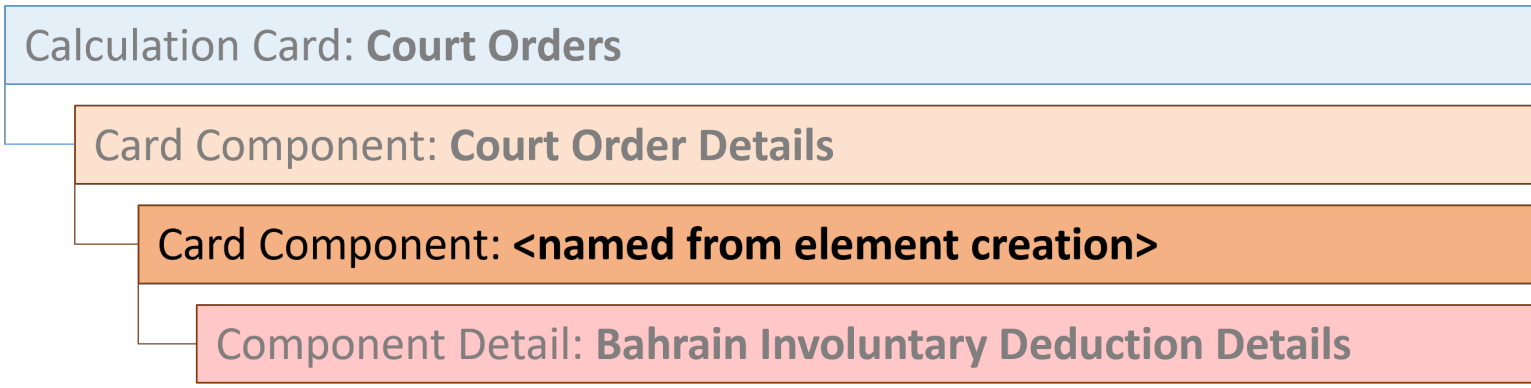
The Court Order Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Court Orders Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders calculation card should be identified by using the same key type used to identify the calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Court Order Details card component. This must be the same or after the EffectiveStartDate on the Court Orders calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Court Orders Details'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. **Note:** Not required when source keys are used.

Card Component Named for the Court Order Element

This card component is a named for the court orders element and is created as a child to the Court Order Details card component.

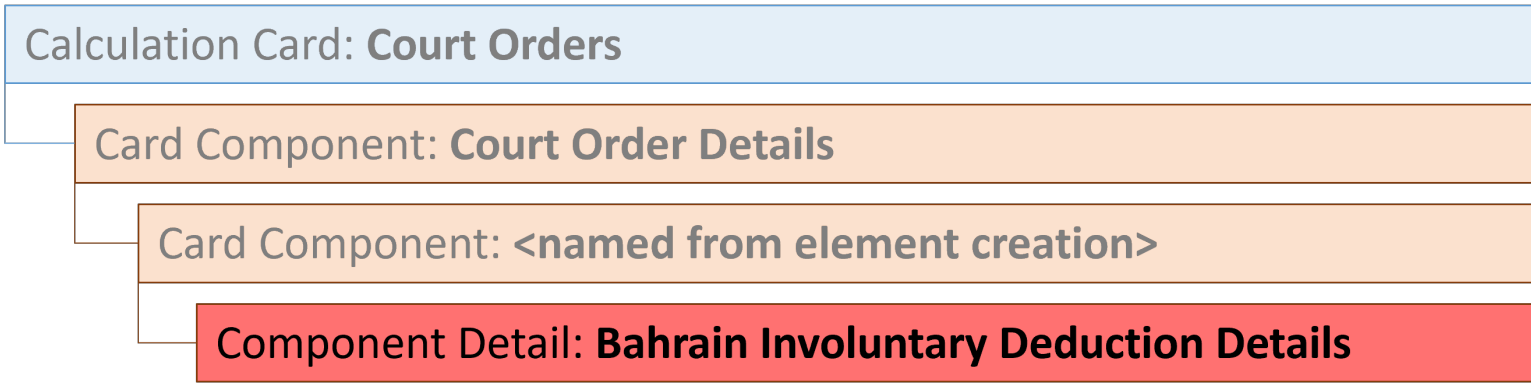


This card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
ParentDirCardCompId (SourceSystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	The parent Court Order Details card component. Use the same key type used to identify the Court Order Details card component.
EffectiveStartDate	N/A	The start date of the card component. This must be the same as the EffectiveStartDate on the Court Orders calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify the name defined when creating the court order element.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.
Context1	N/A	Reference code of the court order.

Component Detail Attributes for Court Orders



Court Orders uses a single flexfield context. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Bahrain Involuntary Deduction Details (ORA_HRX_BH_INVOL_DED_DETAILS)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

To supply third-party details on the Bahrain Involuntary Deduction Details flexfield context, refer to the Cloud Customer Connect topic *BI Publisher Report: Third Party Payees for Involuntary Deduction Cards* for a report that extracts the third-party names and party IDs for the country, legislation and payee type specified.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent card component should be referenced using the same key type used to identify the parent record. This is the card component that is named after the Court Order element. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parentcard component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context. Supply **ORA_HRX_SA_INVOL_DED_DETAILS**. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Court Orders for Bahrain
- Example of Creating Court Orders for Bahrain
- Guidelines for Loading Flexfield Data

Example of Creating Court Orders for Bahrain

This example creates a Court Orders card for employee assignment E3714.

The Bahrain Involuntary Deduction Details flexfield context is populated with these values

Flexfield Segment	Value
Amount	1000
Date of Issue	8 August 2022
Order Amount Payee Details	BH Court Order Payee

The CalculationCard.dat file is used to upload Court Orders calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
AssignmentNumber|EffectiveStartDate
MERGE|CalculationCard|VISION|E3714_CO|BH LDG|Court Orders|E3714|2022/08/08

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
DirCardCompDefName|EffectiveStartDate|Context1|ParentDirCardCompId(SourceSystemId)
MERGE|CardComponent|VISION|E3714_COD|BH LDG|E3714_CO|Court Order Details|2022/08/08||
MERGE|CardComponent|VISION|E3714_BHCO|BH LDG|E3714_CO|PM BH Court Order|2022/08/08|7777|E3714_COD

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|DirCardCompDefName|EffectiveStartDate|DirInformationCategory|
FLEX:Deduction Developer DF|oraHrxBhAmount(Deduction Developer DF=ORA_HRX_BH_INVOL_DED_DETAILS)|
oraHrxBhDateOfIssue(Deduction Developer DF=ORA_HRX_BH_INVOL_DED_DETAILS)|
oraHrxBhThirdPartyPayee_Display(Deduction Developer DF=ORA_HRX_BH_INVOL_DED_DETAILS)
MERGE|ComponentDetail|VISION|E3714_BHCO_ID_DET|BH LDG|E3714_BHCO|PM BH Court Order|2022/08/08|
ORA_HRX_BH_INVOL_DED_DETAILS|ORA_HRX_BH_INVOL_DED_DETAILS|1000.00|2022/08/08|BH Court Order Payee
```

29 Loading Payroll Localization Data for Kuwait

Employee Social Insurance

Overview of Employee Social Insurance Details for Kuwait

The Employee Social Insurance Details card stores all information required to accurately compute social insurance contributions, such as employee details, annuity information and reference salary.

One Employee Social Insurance Details card must be created for every Kuwait employee and Tax Reporting Unit (TRU) that the employee belongs to.

When the product license is set to Payroll or Payroll Interface, Employee Social Insurance Details cards are automatically created when a new employee is entered using the New Hire task with a set of values specified at the person level, such as citizenship and social insurance information. The Employee Social Insurance Details card component is also auto-generated. You only populate the component details if you need to supply override values. There may be cases where this information needs to be loaded in bulk:

During data migration:

Employee Social Insurance information must be uploaded into Oracle Fusion Payroll, to ensure that contributions are calculated correctly. If HCM Data Loader is used to migrate employee records, a default Employee Social Insurance card is automatically created. In most cases, the default won't reflect the employee's actual Social Insurance information, and therefore the card must be updated.

Ongoing bulk updates:

When you bulk load New Hire information, a default Employee Social Insurance card may be automatically generated (if the new hire records are created through HCM Data Loader or the interface with Taleo). In this case, you need to update the default card with the correct social insurance information.

It is recommended to have a good understanding of the Employee Social Insurance card and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS (Kuwait): Payroll Implementation and Functional Considerations (Document ID 1663730.1).

Employee Social Insurance Calculation Card Record Types

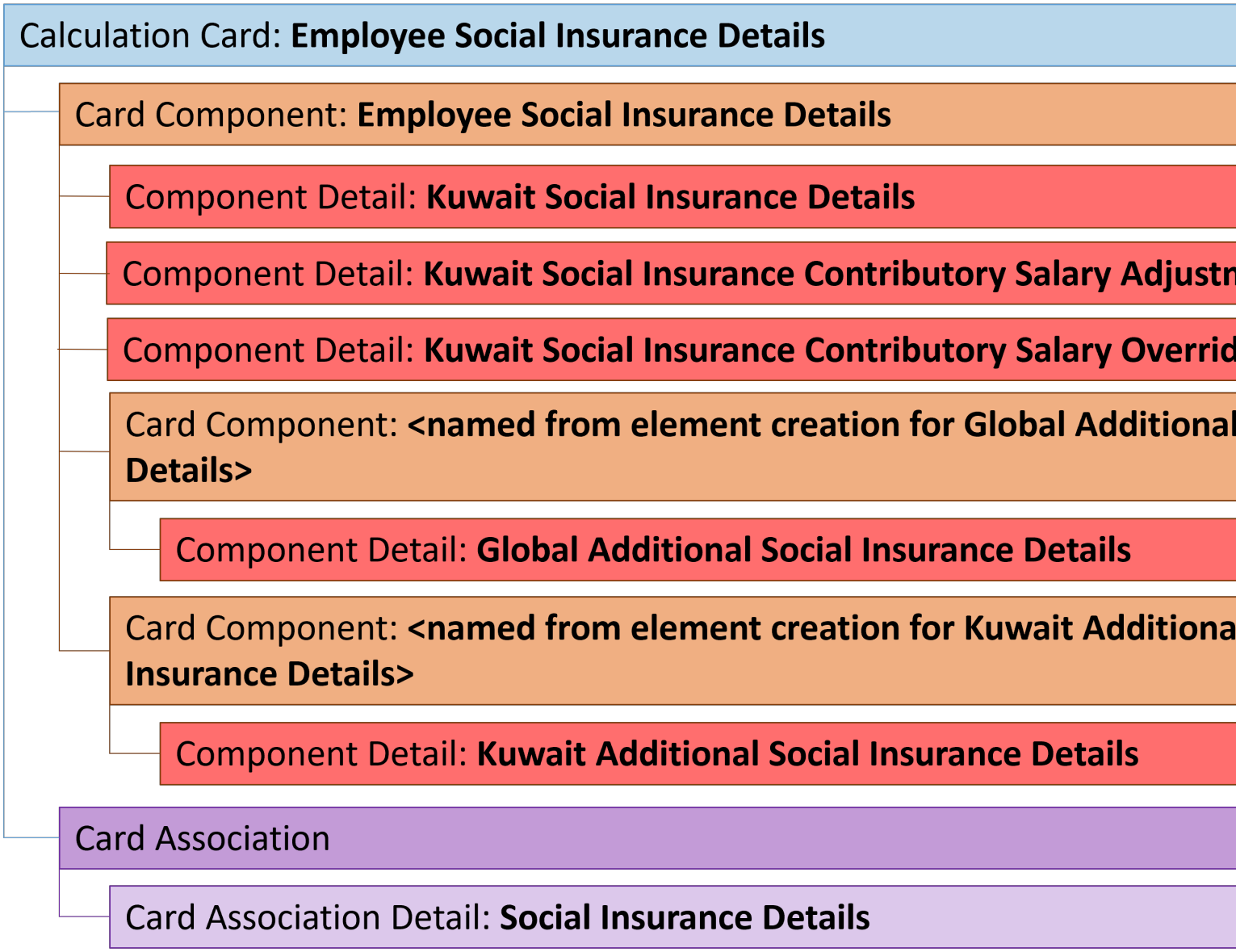
The Employee Social Insurance card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Employee Social Insurance utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card comonoent.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Employee Social Insurance Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Employee Social Insurance are described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Updating Employee Social Insurance Details for Kuwait
- Guidelines for Creating Global Additional Social Insurance Details for Kuwait
- Guidelines for Creating Kuwait Additional Social Insurance Details
- Guidelines for Associating Employee Social Insurance Details Cards for Kuwait
- Example of Updating Employee Social Insurances Details for Kuwait
- Guidelines for Loading Flexfield Data

Guidelines for Updating Employee Social Insurance Details for Kuwait

You must have one Calculation Card record for every Kuwait employee you are maintaining Social Insurance Detail.

Even if you are updating an existing Employee Social Insurance Details card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

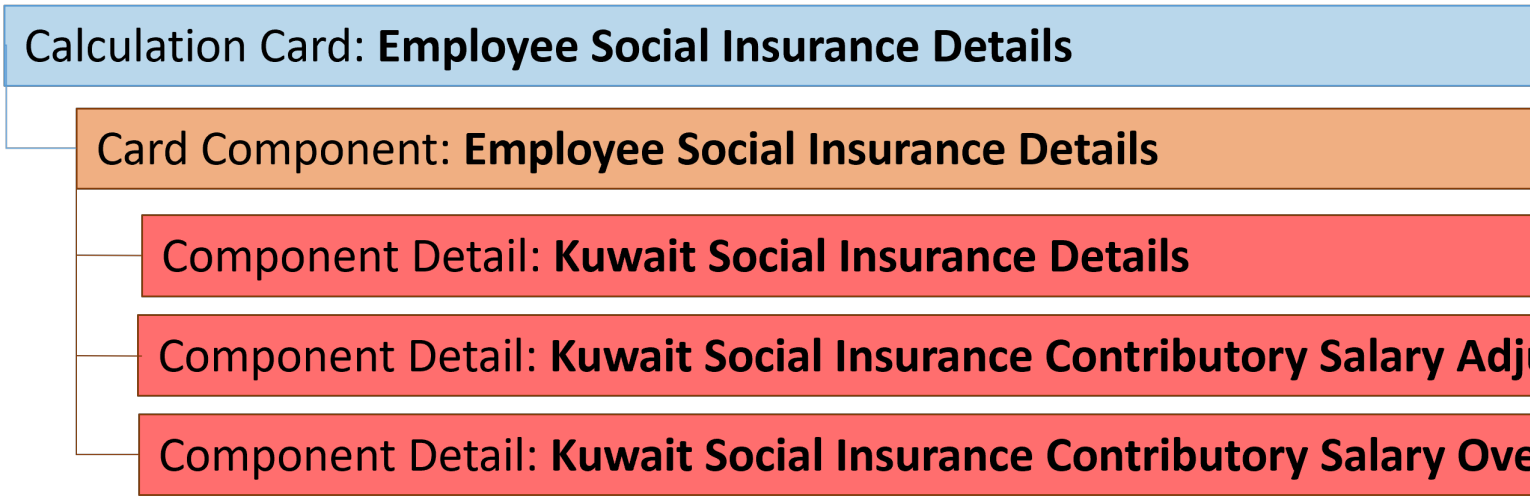
Employee Social Insurance Details Calculation Card Attributes

The Employee Social Insurance Details calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Employee Social Insurance Details calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Employee Social Insurance Details'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Employee Social Insurance Details Card Component

The Employee Social Insurance Details card component captures social insurance related information. It is created for you if the Employee Social Insurance Details card is autogenerated. If you need to update the defaulted social insurance information, provide an Employee Social Insurance Details card component and related Component Detail.



Card Component Attributes for Employee Social Insurance Details Information

The Employee Social Insurance Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee Social Insurance Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Social Insurance Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Employee Social Insurance Details card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Social Insurance Details calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Employee Social Insurance Details'.
ComponentSequence	N/A	Specify '1'

Component Detail Attributes for Employee Social Insurance Details

Employee Social Insurance Details uses three flexfield contexts. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Kuwait Social Insurance Details (HRX_KW_SI_PREL)
- Kuwait Social Insurance Contributory Salary Adjustment (HRX_KW_SI_CONT_SAL_ADJ_PREL)
- Kuwait Social Insurance Contributory Salary Override (HRX_KW_SI_CONT_SAL_OVR_PREL)

Only the Kuwait Social Insurance Details context is required. The others are optional.

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Employee Social Insurance Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Social Insurance Details card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
DirInformationCategory	N/A	The code for the flexfield context, such as '**ORA_HRX_QA_SI_PREL**'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Overview of Employee Social Insurance Details for Kuwait
- Example of Updating Employee Social Insurances Details for Kuwait

Example of Updating Employee Social Insurances Details for Kuwait

These examples update or maintain an auto-generated Employee Social Insurance Details card.

User keys are used to identify the card and card component as the source keys may not be known for auto-generated cards.

Use the CalculationCard.dat file to update Employee Social Insurance Details card component information with HCM Data Loader.

Example of Updating the Auto-Generated Employee Social Insurance Details Card Component

This example updates the existing Employee Social Insurance Details card component for an auto-generated Employee Social Insurance Details card.

The Kuwait Social Insurance Details flexfield context is updated, setting these values

Flexfield Segment	Value
Citizenship	Kuwait
Registered for Social Insurance	Yes
Gratuity Contribution Start Date	1 July 2022

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|KW LDG|Employee Social Insurance Details|2022/01/01|E1616|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|KW LDG|Employee Social Insurance Details|2022/01/01|E1616|1|Employee Social Insurance
 Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|Assignment
 Number|CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction
```

```
 Developer DF|_CITIZENSHIP(Deduction Developer DF=HRX_KW_SI_PREL)|_REGISTERED_FOR_SI(Deduction Developer
 DF=HRX_KW_SI_PREL)|gratuityContributionStart(Deduction Developer DF=HRX_KW_SI_PREL)
MERGE|ComponentDetail|KW LDG|Employee Social Insurance Details|2022/01/01|E1616|1|Employee Social Insurance
 Details|1|HRX_KW_SI_PREL|HRX_KW_SI_PREL|KW|Y|2022/01/01
```

Example of Updating Kuwait Social Insurance Contributory Salary Adjustment

This example populates the Kuwait Social Insurance Contributory Salary Adjustment flexfield context for an auto-generated Employee Social Insurance Details card.

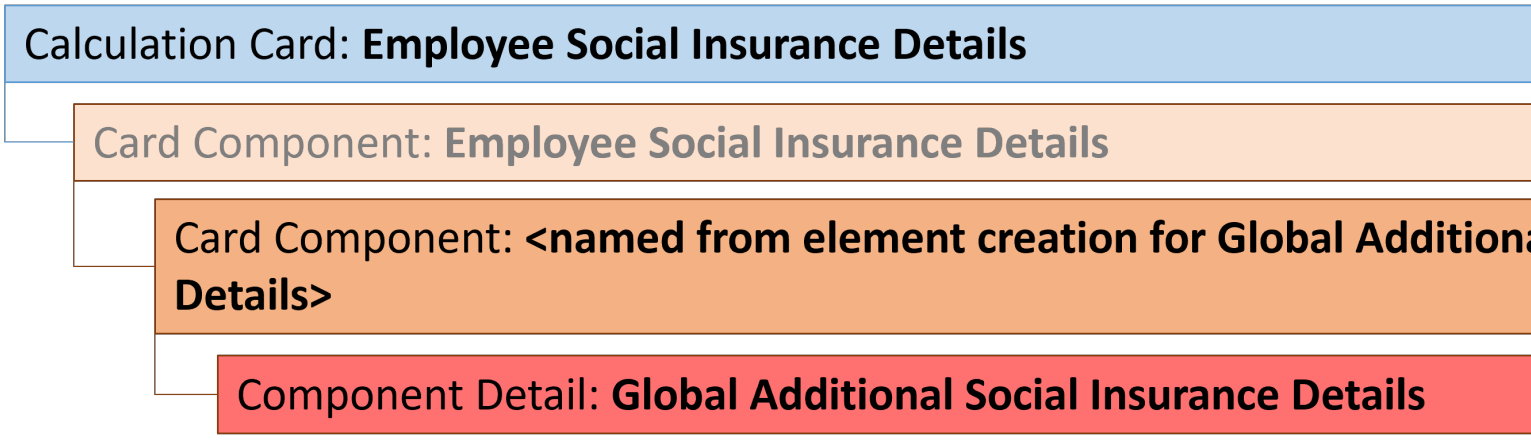
The Adjustment Amount is set to 5000.00.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|KW LDG|Employee Social Insurance Details|2022/07/01|E4321|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|KW LDG|Employee Social Insurance Details|2022/07/01|E4321|1|Employee Social Insurance
 Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|Assignment
 Number|CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer
 DF|_CONTRIBUTORY_SALARY_ADJUSTMENT(Deduction Developer DF=HRX_KW_SI_CONT_SAL_ADJ_PREL)
MERGE|ComponentDetail|KW LDG|Employee Social Insurance Details|2022/07/01|E4321|1|Employee Social Insurance
 Details|1|HRX_KW_SI_CONT_SAL_ADJ_PREL|HRX_KW_SI_CONT_SAL_ADJ_PREL|5000.00
```

Guidelines for Creating Global Additional Social Insurance Details for Kuwait

The Global Additional Social Insurance Details card component records global additional social insurance related information.

The card component is named by the element created for Global Additional Social Insurance Details.



Supply the Global Additional Social Insurance Details card component as a child of the Employee Social Insurance Details card component, with a component detail record for the Global Additional Social Insurance Details flexfield segment attributes.

Card Component Attributes for Global Additional Social Insurance Details

The Global Additional Social Insurance Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Global Additional Social Insurance Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Social Insurance Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
ParentDirCardCompId (SourceSystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	The parent Employee Social Insurance Details card component should be identified using the same key type used to identify the card component.
EffectiveStartDate	N/A	The start date of the Global Additional Social Insurance Details card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Social Insurance Details calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify the name defined when creating the Global Additional Social Insurance Details element.
ComponentSequence	N/A	Specify '1'.
Context1	N/A	The insurance type for the Global Additional Social Insurance Details.

Component Detail Attributes for Global Additional Social Insurance Details

Global Additional Social Insurance Details uses one flexfield context. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Global Additional Social Insurance Details (ORA_PAY_ADDITIONAL_SI_DETAILS)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Global Additional Social Insurance Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Global Additional Social Insurance Details card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'ORA_PAY_ADDITIONAL_SI_DETAILS'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

Example of Creating Global Additional Social Insurance Details for Kuwait

This example creates the Global Additional Social Insurance Details card component for an auto-generated Employee Social Insurance Details card. User keys are provided to identify the card.

The Partial Deduction Arrears Rule is being set to 'Do not take a partial deduction or create arrears'. This flexfield segment is validated by the lookup type PAY_TMPLT_DEDN_ARREAR and the lookup code is supplied to the _PARTIAL_DEDUCTION_ARREARS_RULE(Deduction Developer DF=ORA_PAY_ADDITIONAL_SI_DETAILS) attribute.

Use the CalculationCard.dat file to upload Global Additional Social Insurance Details data.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|EffectiveStartDate|
CardSequence
MERGE|CalculationCard|KW LDG|Employee Social Insurance Details|E4321|2022/07/01|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|DirCardCompDefName|EffectiveStartDate|
AssignmentNumber|CardSequence|Context1|ComponentSequence|ParentDirCardCompDefName|ParentComponentSequence
MERGE|CardComponent|KW LDG|Employee Social Insurance Details|Employee Social Insurance Details|2022/07/01|
E4321|1||1||
MERGE|CardComponent|KW LDG|Employee Social Insurance Details|KW Global Additional SI Details|2022/07/01|
E4321|1|Basic insurance contributions|1|Employee Social Insurance Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|DirCardCompDefName|
EffectiveStartDate|AssignmentNumber|CardSequence|ComponentSequence|DirInformationCategory|FLEX:Deduction
 Developer DF|_PARTIAL_DEDUCTION_ARREARS_RULE(Deduction Developer DF=ORA_PAY_ADDITIONAL_SI_DETAILS)
MERGE|ComponentDetail|KW LDG|Employee Social Insurance Details|KW Global Additional SI Details|2022/07/01|
E4321|1|1|ORA_PAY_ADDITIONAL_SI_DETAILS|ORA_PAY_ADDITIONAL_SI_DETAILS|NPNA
```

Guidelines for Creating Kuwait Additional Social Insurance Details

The Kuwait Additional Social Insurance Details card component records additional social insurance information related to Kuwait.

The card component is named by the element created for Kuwait Additional Social Insurance Details.

Calculation Card: **Employee Social Insurance Details**

Card Component: **Employee Social Insurance Details**

Card Component: **<named from element creation for Kuwait Addition Insurance Details>**

Component Detail: **Kuwait Additional Social Insurance Details**

Supply the Kuwait Additional Social Insurance Details card component as a child of the Employee Social Insurance Details card component, with a component detail record for the Kuwait Additional Social Insurance Details flexfield segment attributes.

Card Component Attributes for Kuwait Additional Social Insurance Details

The Kuwait Additional Social Insurance Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Kuwait Additional Social Insurance Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Social Insurance Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
ParentDirCardCompId (SourceSystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	The parent Employee Social Insurance Details card component should be identified using the same key type used to identify the card component.
EffectiveStartDate	N/A	The start date of the Kuwait Additional Social Insurance Details card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Social Insurance Details calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify the name defined when creating the Kuwait Additional Social Insurance Details element.
ComponentSequence	N/A	Specify '1'.
Context1	N/A	The insurance type for the Kuwait Additional Social Insurance Details.

Component Detail Attributes for Kuwait Additional Social Insurance Details

Kuwait Additional Social Insurance Details uses one flexfield context. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Kuwait Additional Social Insurance Details (ORA_HRX_KW_ADDITIONAL_SI_DETAILS)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Kuwait Additional Social Insurance Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Kuwait Additional Social Insurance Details card component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'ORA_HRX_KW_ADDITIONAL_SI_DETAILS'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Example of Creating Kuwait Additional Social Insurance Details

This example creates the Kuwait Additional Social Insurance Details card component for an auto-generated Employee Social Insurance Details card. User keys are provided to identify the card.

The Monthly Payment is being set to 500.00, the Total Debt is set to 1000.00.

Use the CalculationCard.dat file to upload Kuwait Additional Social Insurance Details data.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|EffectiveStartDate|
CardSequence
MERGE|CalculationCard|KW LDG|Employee Social Insurance Details|E4321|2022/07/01|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|EffectiveStartDate|
CardSequence|DirCardCompDefName|ComponentSequence|Context1|ParentDirCardCompDefName|ParentComponentSequence

MERGE|CardComponent|KW LDG|Employee Social Insurance Details|E4321|2022/07/01|1|Employee Social Insurance
 Details|1|||
MERGE|CardComponent|KW LDG|Employee Social Insurance Details|E4321|2022/07/01|1|KW Kuwait Additional SI
 Details|1|Basic insurance contributions|Employee Social Insurance Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|AssignmentNumber|EffectiveStartDate|
CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
_MONTHLY_PAYMENT(Deduction Developer DF=ORA_HRX_KW_ADDITIONAL_SI_DETAILS)|_TOTAL_DEBT(Deduction Developer
 DF=ORA_HRX_KW_ADDITIONAL_SI_DETAILS)
MERGE|ComponentDetail|KW LDG|Employee Social Insurance Details|E4321|2022/07/01|1|KW Kuwait Additional SI
 Details|1|ORA_HRX_KW_ADDITIONAL_SI_DETAILS|ORA_HRX_KW_ADDITIONAL_SI_DETAILS|500.00|1000.00
```

Guidelines for Associating Employee Social Insurance Details Cards for Kuwait

The card association record associates the Employee Social Insurance Details calculation card with the employee's tax reporting unit.

Calculation Card: **Employee Social Insurance Details**

Card Component: **Employee Social Insurance Details**

Card Association

Card Association Detail: **Employee Social Insurance Details**

The associated tax reporting unit is defined in the card association. The card association detail allows the Employee Social Insurance Details card component to be associated with the employee's assignments.

Card Association Attributes for Employee Social Insurance Details

The Card Association record type associates the Employee Social Insurance Details card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Employee Social Insurance Details card association. For new card associations, supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Social Insurance Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Social Insurance Details calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Employee Social Insurace Details calculation card. If source keys are used

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Employee Social Insurance Details

The card association detail associates the Employee Social Insurance Details card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations, supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Thecard component this association is for. Supply either the source system ID value or user key attributes to identify the Employee Social InsuranceDetails card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of Employee Social Insurance Details for Kuwait

Example of Loading Employee Social Insurance Details Card Associations for Kuwait

This example creates the tax reporting unit association for the existing Employee Social Insurance Details card belonging to employee assignment E4321.

User keys are provided to reference the existing Employee Social Insurance Details card and Employee Social Insurance Details card component on the assumption they were auto generated and therefore the source key values are unknown.

The CalculationCard.dat file is used to upload the Employee Social Insurance Details card associations with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|KW LDG|Employee Social Insurance Details|2022/07/01|E4321|1

METADATA|CardAssociation|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|TaxReportingUnitName
MERGE|CardAssociation|KW LDG|Employee Social Insurance Details|2022/07/01|E4321|1|KW Tax Reporting Unit

METADATA|CardAssociationDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|
AssignmentNumber|CardSequence|TaxReportingUnitName|DirCardCompDefName|ComponentSequence|
AssociationAssignmentNumber
MERGE|CardAssociationDetail|KW LDG|Employee Social Insurance Details|2022/07/01|E4321|1|KW Tax Reporting
 Unit|Employee Social Insurance Details|1|E4321
```

When associating an auto-generated card you don't need to include the CardComponent unless also making changes to its values.

Example of Creating Employee Social Insurance Details for Kuwait

This example creates an Employee Social Insurance Details card with associations for employee assignment E4321.

If your Employee Social Insurance Details cards are not auto-generated it is recommended that you use source keys to create the card.

The Kuwait Social Insurance Details flexfield context is populated with these values

Flexfield Segment	Value
Citizenship	Kuwait
Registered for Social Insurance	Yes
Gratuity Contribution Start Date	1 July 2022

The CalculationCard.dat file is used to upload Employee Social Insurance Details calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|ESI_E4321|KW LDG|Employee Social Insurance Details|2022/07/01|E4321

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|ESI_E4321_ESID|KW LDG|ESI_E4321|2022/07/01|Employee Social Insurance Details

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|FLEX:Deduction
 Developer DF|_CITIZENSHIP(Deduction Developer DF=HRX_KW_SI_PREL)|_REGISTERED_FOR_SI(Deduction Developer
 DF=HRX_KW_SI_PREL)|gratuityContributionStart(Deduction Developer DF=HRX_KW_SI_PREL)
MERGE|ComponentDetail|VISION|ESI_E4321_FLX|KW LDG|ESI_E4321_ESID|2022/07/01|Employee Social Insurance
 Details|HRX_KW_SI_PREL|HRX_KW_SI_PREL|KW|Y|2022/07/01

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|ESI_E4321_ASS|KW LDG|ESI_E4321|2022/07/01|KW Tax Reporting Unit

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirRepCardId(SourceSystemId)|
DirCardCompId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|ESI_E4321_ASD|ESI_E4321_ASS|ESI_E4321_ESID|2022/07/01|E4321
```

Employee Gratuity Details

Overview of Employee Gratuity Details for Kuwait

The Employee Gratuity Details card stores all information required to calculate and process the gratuity payment, such as gratuity override amount and payment date.

The Employee Gratuity Details card is created automatically upon employee termination and the gratuity amount calculated. When the gratuity card is created, the Gratuity Details card component, Kuwait Employee Gratuity Details component detail, and card association are also created.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, Employee Gratuity Details cards are automatically created upon employee termination. There might be cases where this information needs to be loaded in bulk:

During data migration:

Employee Gratuity Details information must exist into Oracle Fusion Payroll to ensure that values are calculated correctly. If HCM Data Loader is used to migrated employee termination records, a default Employee Gratuity Details card is automatically created. For historical reasons, you might wish to upload gratuity calculation cards to record the Last Gratuity Payment Date for a previous employment.

Ongoing bulk updates:

- When you bulk load employee terminations, a default Employee Gratuity Details calculation card might be generated automatically (if the terminations records are created using HCM Data Loader). In this case, you need to update the default card with the correct gratuity information. Also if needed, create a component detail record to store a gratuity override amount.

It is recommended to have a good understanding of the Employee Gratuity Details card and the information it contains before attempting mass upload as it has a direct impact on calculations and reporting. For further information, see Oracle Fusion HRMS (Kuwait): Payroll Implementation and Functional Considerations (Document ID 1663730.1).

Employee Gratuity Details Calculation Card Record Types

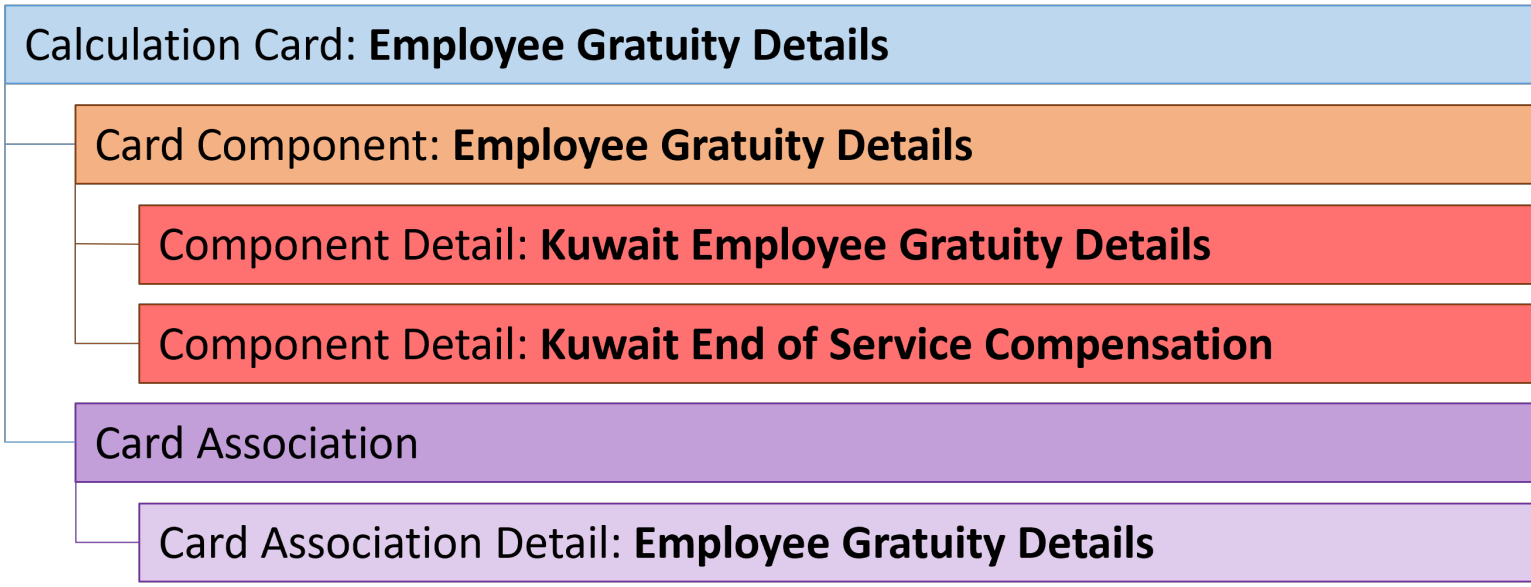
The Employee Gratuity Details card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Employee Gratuity Details uses these record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Employee Gratuity Details Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Employee Gratuity Details is described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Employee Gratuity Details for Kuwait
- Guidelines for Associating Employee Gratuity Details Cards for Kuwait
- Example of Updating Employee Gratuity Details for Kuwait
- Guidelines for Loading Flexfield Data

Guidelines for Loading Employee Gratuity Details for Kuwait

You must have one Calculation Card record for every Kuwait employee you are maintaining Gratuity Details for.

Even if you are updating an existing Employee Gratuity Details card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Employee Gratuity Details Calculation Card Attributes

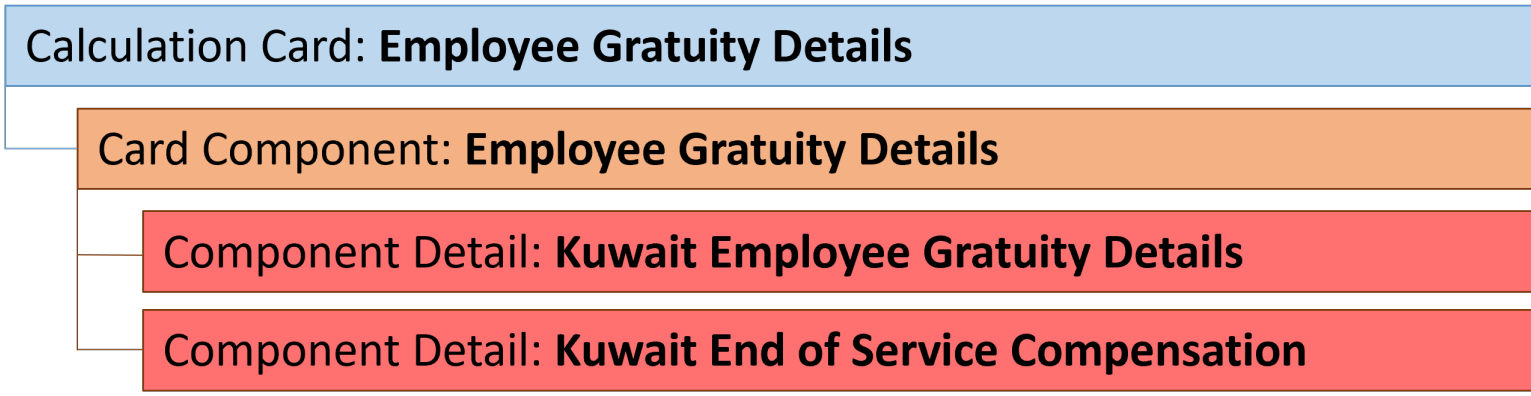
The Employee Gratuity Details calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Employee Gratuity Details calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Employee Gratuity Details'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's termination date.
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Employee Gratuity Details Card Component

The Employee Gratuity Details card component captures social insurance-related information. It is created for you if the Employee Gratuity Details card is autogenerated.

If you need to update the defaulted gratuity information, provide an Employee Gratuity Details card component with a Component Detail record for each flexfield context relevant to the employee.



Card Component Attributes for Gratuity Details

The Employee Gratuity Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee Gratuity Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Gratuity Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Employee Gratuity Details card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Gratuity Details calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Employee Gratuity Details'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		the same card component definition exist on the same card. Not required when source keys are used.

Component Detail Attributes for Employee Gratuity Details

Employee Gratuity Details uses one flexfield context. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Kuwait Employee Gratuity Details (HRX_KW_GRATUITY_PREL)
- Kuwait End of Service Compensation (HRX_KW_GRATUITY_EOS_COMP)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Employee Gratuity Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Gratuity Details card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context. For example HRX_KW_GRATUITY_PREL.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Employee Gratuity Details for Kuwait
- Guidelines for Associating Employee Gratuity Details Cards for Kuwait
- Guidelines for Loading Flexfield Data

Example of Updating Employee Gratuity Details for Kuwait

This example updates an auto-generated Gratuity Details card component of the Employee Gratuity Details card.

The Kuwait Employee Gratuity Details flexfield context is populated with these values for employee assignment 4321.

Flexfield Segment	Value
Override Amount	1050.00
Latest Gratuity Payment Date	31 July 2022

The CalculationCard.dat file is used to update Employee Gratuity Details calculation cards with HCM Data Loader.

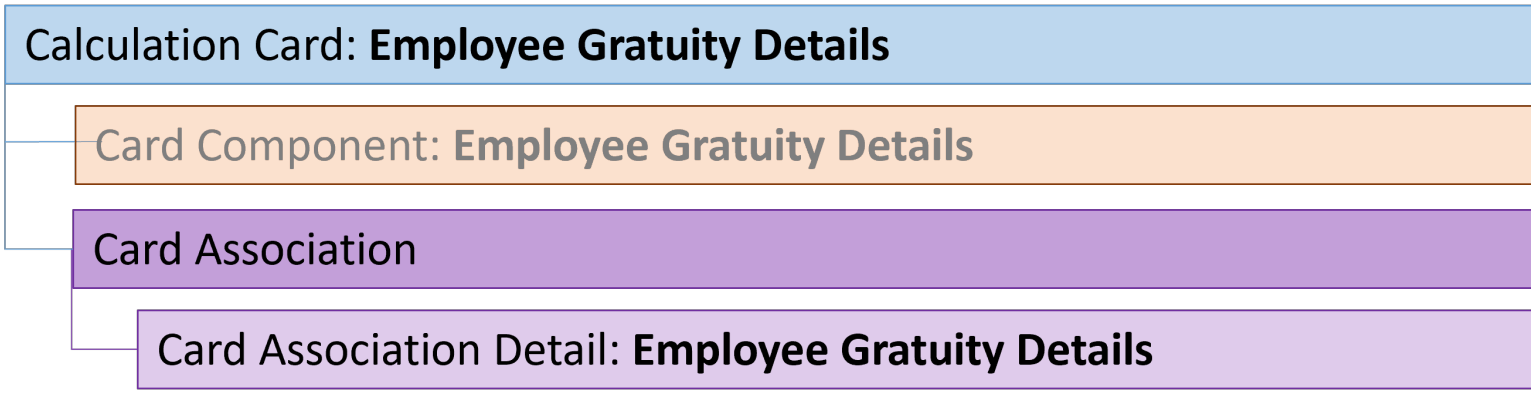
```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber
MERGE|CalculationCard|KW LDG|Employee Gratuity Details|1|2022/07/01|E4321

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|KW LDG|Employee Gratuity Details|1|2022/07/01|E4321|Employee Gratuity Details|11

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
_OVERRIDE_AMOUNT(Deduction Developer DF=HRX_KW_GRATUITY_PREL)|_ORA_HRX_KW_LST_GRAT_PYMT_DATE(Deduction
 Developer DF=HRX_KW_GRATUITY_PREL)
MERGE|ComponentDetail|KW LDG|Employee Gratuity Details|1|2022/07/01|E4321|Employee Gratuity Details|11|
HRX_KW_GRATUITY_PREL|HRX_KW_GRATUITY_PREL|1050.00|2022/07/31
```

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Employee Gratuity Details for Kuwait
- Guidelines for Loading Flexfield Data

Guidelines for Associating Employee Gratuity Details Cards for Kuwait

The card association record associates the Employee Gratuity Details calculation card with the employee's tax reporting unit.



The associated tax reporting unit is defined in the card association. The card association detail allows the Employee Gratuity Details card component to be associated with the employee's assignments.

Card Association Attributes for Employee Gratuity Details

The Card Association record type associates the Employee Gratuity Details card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Employee Gratuity Details card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Gratuity Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Tax Withholding calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Employee Gratuity Details calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Employee Gratuity Details

The card association detail associates the Employee Gratuity Details card component with the payroll assignments for the employee.

If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Thecard component this association is for. Supply either the source system ID value or user key attributes to identify the Employee Gratuity Details card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of Employee Gratuity Details for Kuwait
- Guidelines for Loading Employee Gratuity Details for Kuwait

Example of Loading Employee Gratuity Details Card Associations for Kuwait

This example creates the tax reporting unit association for the existing Employee Gratuity Details card belonging to employee assignment E4321.

User keys are provided to reference the existing Employee Gratuity Details card and Employee Gratuity Details card component on the assumption they were auto generated and therefore the source key values are unknown.

The CalculationCard.dat file is used to upload the Employee Gratuity Details card associations with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber
MERGE|CalculationCard|KW LDG|Employee Gratuity Details|1|2022/07/01|E4321

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|KW LDG|Employee Gratuity Details|1|2022/07/01|E4321|Employee Gratuity Details|11

METADATA|CardAssociation|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|TaxReportingUnitName
MERGE|CardAssociation|KW LDG|Employee Gratuity Details|1|2022/07/01|E4321|KW Legal Entity

METADATA|CardAssociationDetail|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|
EffectiveStartDate|AssignmentNumber|TaxReportingUnitName|DirCardCompDefName|ComponentSequence|
AssociationAssignmentNumber
MERGE|CardAssociationDetail|KW LDG|Employee Gratuity Details|1|2022/07/01|E4321|KW Legal Entity|Gratuity
 Details|11|E4321
```

When associating an auto-generated card you don't need to include the CardComponent unless also making changes to its values.

Example of Creating Employee Gratuity Details for Kuwait

This example creates the Employee Gratuity Details calculation card with associations for employee assignment E3434 on the 1st Jan 2022, for a payment date of 31st Jan 2022.

If your Employee Gratuity Details cards are not auto-generated it is recommended that you use source keys to create the card.

The Kuwait Employee Gratuity Details flexfield context is populated with these values.

Flexfield Segment	Value
Gratuity Override Amount	10000.00

Flexfield Segment	Value
Latest Gratuity Payment Date	31 Jan 2022

The CalculationCard.dat file is used to upload Employee Gratuity Details calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|EGD_E4321|KW LDG|Employee Gratuity Details|2022/01/01|E4321

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|EGD_E4321_CC|KW LDG|EGD_E4321|2022/01/01|Employee Gratuity Details

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|_OVERRIDE_AMOUNT(Deduction Developer DF=HRX_KW_GRATUITY_PREL)|
_ORA_HRX_KW_LST_GRAT_PYMT_DATE(Deduction Developer DF=HRX_KW_GRATUITY_PREL)
MERGE|ComponentDetail|VISION|EGD_E4321_KWG|KW LDG|EGD_E4321_CC|2022/01/01|Employee Gratuity Details|
HRX_KW_GRATUITY_PREL|HRX_KW_GRATUITY_PREL|1000.00|2022/01/31

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|EGD_E4321_A|KW LDG|EGD_E4321|2022/01/01|KW Tax Reporting Unit

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirRepCardId(SourceSystemId)|EffectiveStartDate|DirCardCompId(SourceSystemId)|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|EGD_E4321_AD|KW LDG|EGD_E4321_A|2022/01/01|EGD_E4321_CC|E4321
```

Court Orders

Overview of Court Orders for Kuwait

The Court Orders card stores information for court order deductions, such as amount, frequency, and payee.

Considerations and Prerequisites

Use the Elements task to create Court Order elements. When you create court order elements, a card component definition is generated with the same name as the element. The generated card component definition is then available to use on the predefined Court Orders calculation card.

Unlike Employee Social Insurance Details or Employee Gratuity Details cards, the Court Order card is not generated automatically. You only need to create this card for those employees who are subject to court order deductions.

There may be cases when there is a need to bulk load Court Orders cards:

During data migration:

Employee's Court Orders must bemigrated to the Oracle Payroll Cloud to ensure that appropriate deductions are consider. Using HCM Data Loader you must create the card and all necessary card components and component details.

Ongoing bulk updates:

- New hires may have Court Orders that need to be uploaded in bulk.

It is recommended to have a good understanding of the Court Orders card and the information it contains prior to attempting mass upload. For further information, see Oracle Fusion HRMS (Kuwait): Payroll Implementation and Functional Considerations (Doc ID 1663730.1).

Before bulk loading Court Order information:

- Create eligibility for the Court Order element.
- Create third-party payee and third-party payment methods for the people and organizations the payment of the court order is made to.
- Update the PAY_INVLN_DEDN_ORDER_AMOUNT_PAYEE lookup type with the court order issuing authorities.

Court Order Calculation Card Record Types

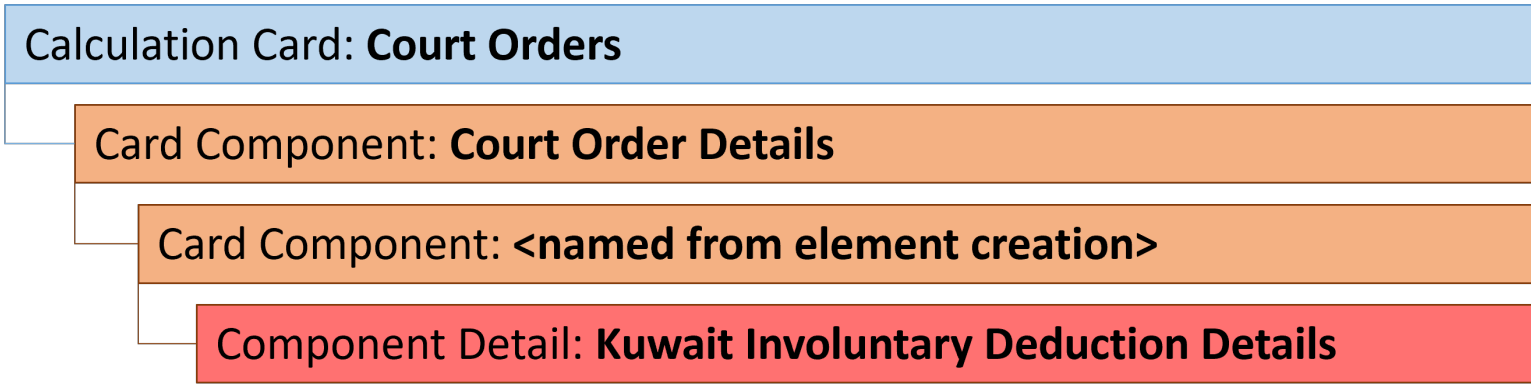
The Court Orders card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Court Orders card utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail

Court Orders Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Court Orders are described in this diagram:



Related Topics

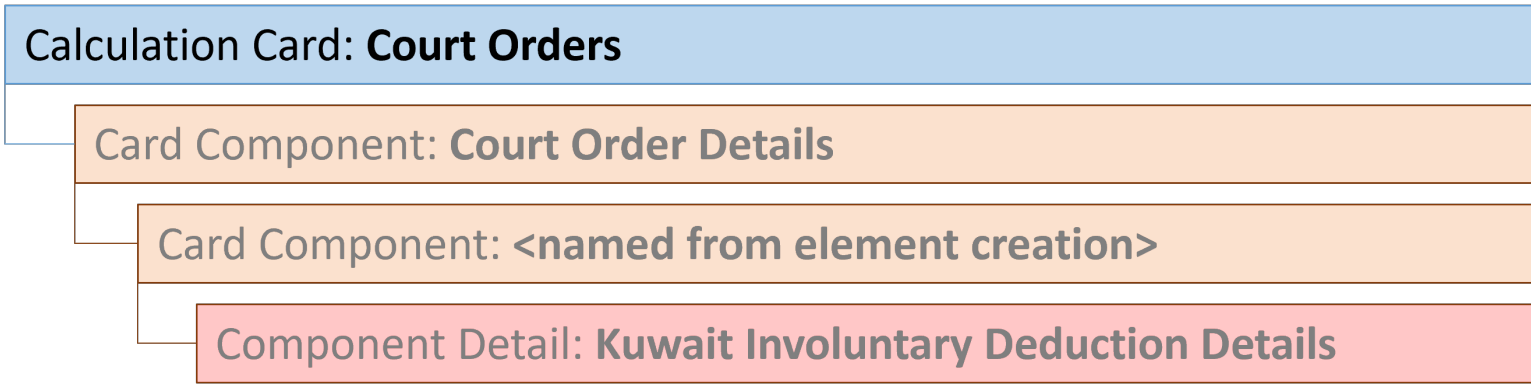
- Guidelines for Loading Calculation Cards
- Guidelines for Loading Court Orders for Kuwait
- Guidelines for Loading Flexfield Data

Guidelines for Loading Court Orders for Kuwait

You should always include the Court Orders calculation card record, even if you're updating an existing Court Orders card.

Even if the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Court Orders Calculation Card Attributes

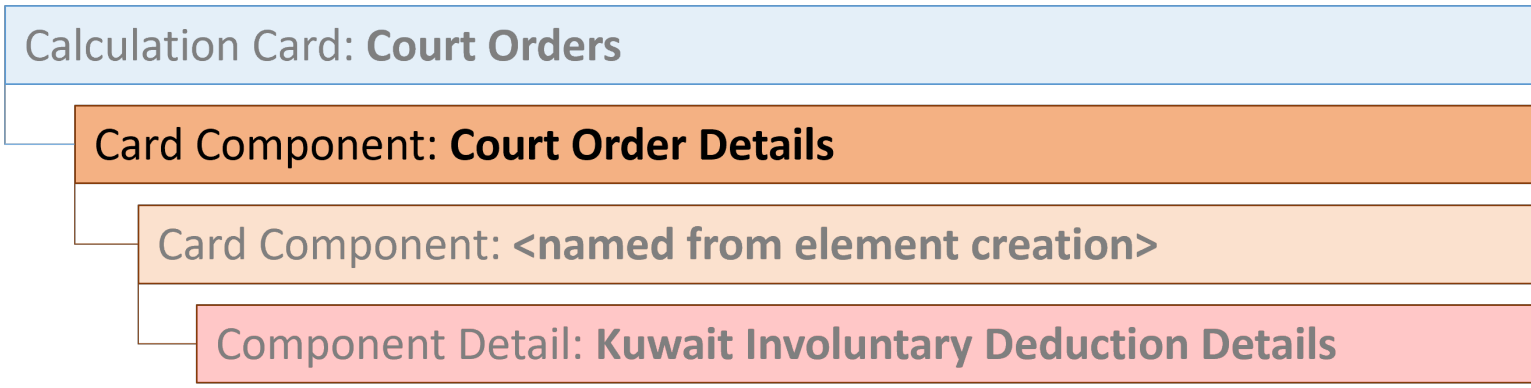


The Court Orders calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the **Court Orders** calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Court Orders'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Court Order Details Card Component

The Court Order Details card component is a predefined card component which acts as a parent to the element specific card component.



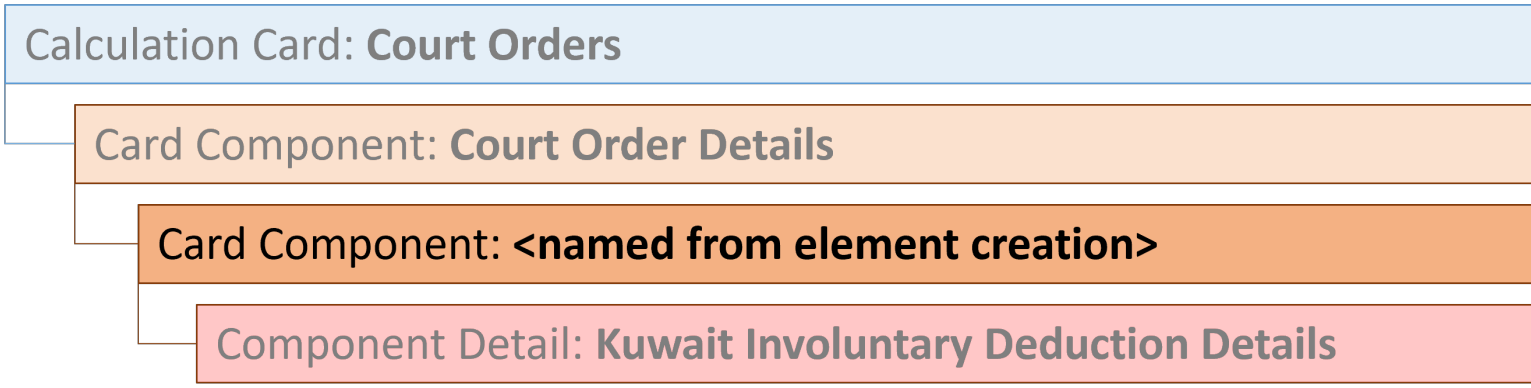
The Court Order Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Court Orders Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Court Order Details card component. This must be the same or after the EffectiveStartDate on the Court Orders calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Court Orders Details'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		the same card. Not required when source keys are used.

Card Component Named for the Court Order Element

This card component is a named for the court orders element and is created as a child to the Court Order Details card component.

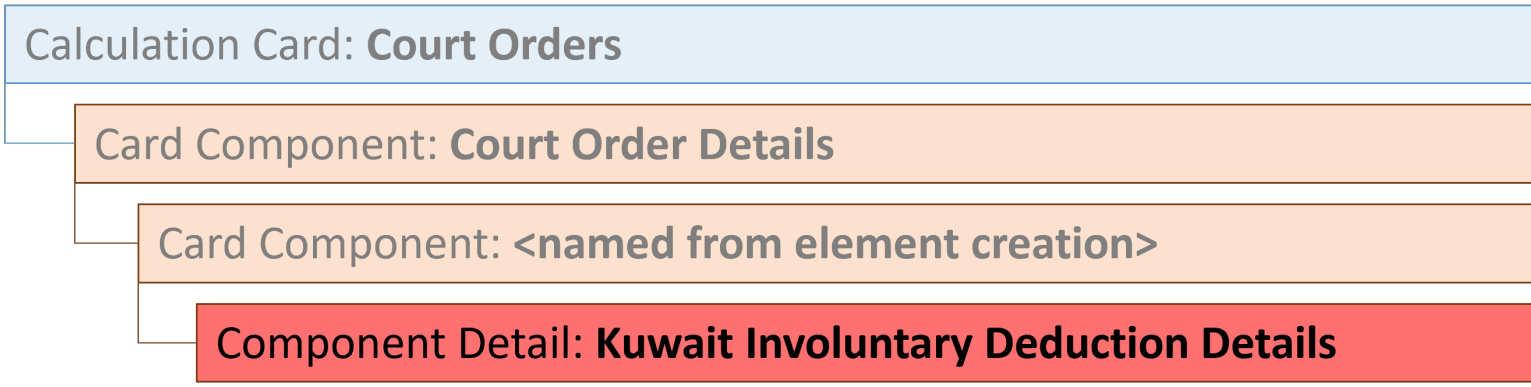


This card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent **Court Orders** calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
ParentDirCardCompId (SourceSystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	The parent Court Order Details card component. Use the same key type used to identify the Court Order Details card component.
EffectiveStartDate	N/A	The start date of the card component. This must be the same as the EffectiveStartDate on the **Court Orders** calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify the name defined when creating the court order element.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.
Context1	N/A	Reference code of the court order.

Component Detail Attributes for Court Orders



Court Orders uses a single flexfield context. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Kuwait Involuntary Deduction Details (ORA_HRX_KW_INVOL_DED_DETAILS)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

To supply third-party details on the Kuwait Involuntary Deduction Details flexfield context, refer to the Cloud Customer Connect topic *BI Publisher Report: Third Party Payees for Involuntary Deduction Cards* for a report that extracts the third-party names and party IDs for the country, legislation and payee type specified.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent card component should be referenced using the same key type used to identify the parent record. This is the card component that is named after the Court Order element. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parentcard component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context. Supply **ORA_HRX_SA_INVOL_DED_DETAILS**. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Example of Creating Court Orders for Kuwait
- Guidelines for Loading Flexfield Data
- Overview of Court Orders for Kuwait

Example of Creating Court Orders for Kuwait

This example creates a Court Orders card for employee assignment E4321.

The Kuwait Involuntary Deduction Details flexfield context is populated with these values:

Flexfield Segment	Value
Amount	1000
Date of Issue	1 Jan 2022
Order Amount Payee Details	KW Court Order Payee

The CalculationCard.dat file is used to upload Court Orders calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
AssignmentNumber|EffectiveStartDate
MERGE|CalculationCard|VISION|COE4321|KW LDG|Court Orders|E4321|2022/01/01

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
DirCardCompDefName|EffectiveStartDate|Context1|ParentDirCardCompId(SourceSystemId)
MERGE|CardComponent|VISION|COE4321COD|KW LDG|COE4321|Court Order Details|2022/01/01||
MERGE|CardComponent|VISION|COE4321KCO|KW LDG|COE4321|KW Court Order|2022/01/01|Court Order Reference|
COE4321COD

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|DirCardCompDefName|EffectiveStartDate|DirInformationCategory|FLEX:Deduction
 Developer DF|amount(Deduction Developer DF=ORA_HRX_KW_INVOL_DED_DETAILS)|dateOfIssue(Deduction
 Developer DF=ORA_HRX_KW_INVOL_DED_DETAILS)|orderAmountPayeeDetails_Display(Deduction Developer
 DF=ORA_HRX_KW_INVOL_DED_DETAILS)
MERGE|ComponentDetail|VISION|COE4321IND|KW LDG|COE4321KCO|KW Court Order|2022/01/01|
ORA_HRX_KW_INVOL_DED_DETAILS|ORA_HRX_KW_INVOL_DED_DETAILS|1000.00|2022/01/01|KW Court Order Payee
```

30 Loading Payroll Localization Data for Qatar

Employee Gratuity Details

Overview of Employee Gratuity Details for Qatar

The Employee Gratuity Details card stores all information required to calculate and process the gratuity payment, such as gratuity override amount and payment date.

The Employee Gratuity Details card is created automatically upon employee termination and the gratuity amount calculated. When the gratuity card is created, the Gratuity Details cardcomponent, Qatar Employee Gratuity Details component detail and card association are also created.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, Employee Gratuity Details cards are automatically created upon employee termination. There may be cases where this information needs to be loaded in bulk:

During data migration:

Employee Gratuity Details information must exist into Oracle Fusion Payroll to ensure that values are calculated correctly. If HCM Data Loader is used to migrated employee termination records, a default Employee Gratuity Details card is automatically created. For historical reasons you may wish to upload gratuity calculation cards to records the Last Gratuity Payment Date for a previous employment.

Ongoing bulk updates:

- When you bulk load employee terminations a default Employee Gratuity Details calculation card may be generated automatically (if the terminations records are created using HCM Data Loader). In this case, you need to update the default card with the correct gratuity information and, if needed, create a component detail record to store a gratuity override amount

It is recommended to have a good understanding of the Employee Gratuity Details card and the information it contains prior to attempting mass upload as it has a direct impact on calculations and reporting. For further information, see Oracle Fusion HRMS (Qatar): Payroll Implementation and Functional Considerations (Document ID 2329792.1).

Employee Gratuity Details Calculation Card Record Types

The Employee Gratuity Details card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

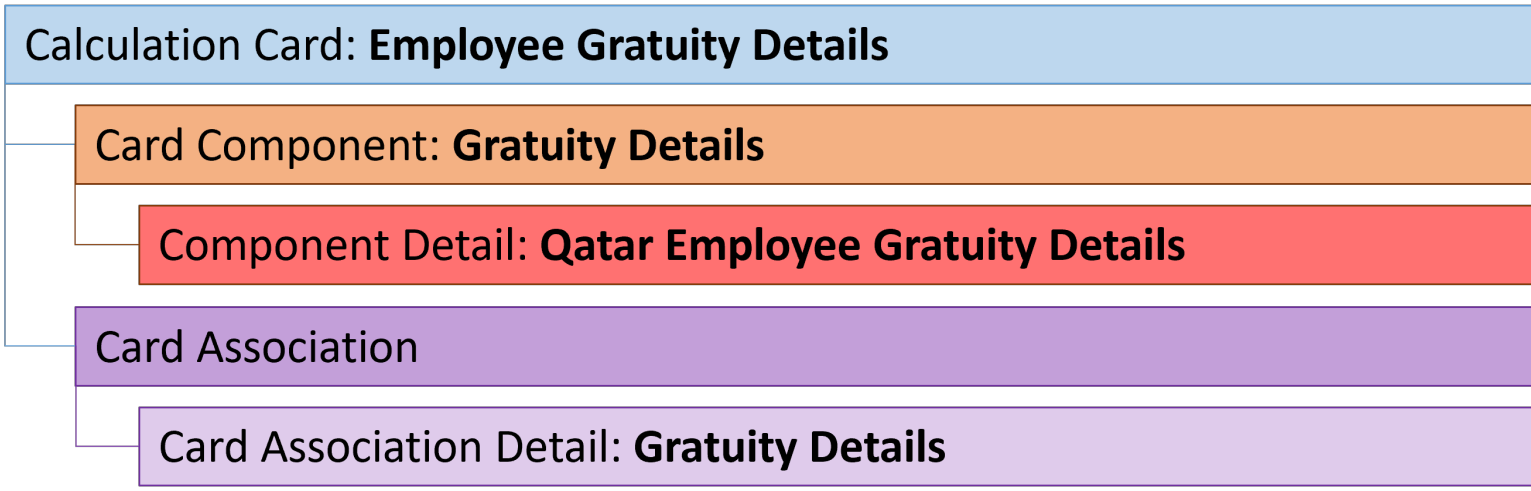
The Employee Gratuity Details utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard

Component	Functional Description	File Discriminator
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Employee Gratuity Details Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Employee Gratuity Details are described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Employee Gratuity Details for Qatar
- Guidelines for Associating Employee Gratuity Details Cards for Qatar
- Example of Updating Employee Gratuity Details for Qatar
- Guidelines for Loading Flexfield Data

Guidelines for Loading Employee Gratuity Details for Qatar

You must have one Calculation Card record for every Qatar employee you are maintaining Gratuity Details for.

Even if you are updating an existing Employee Gratuity Details card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Employee Gratuity Details Calculation Card Attributes

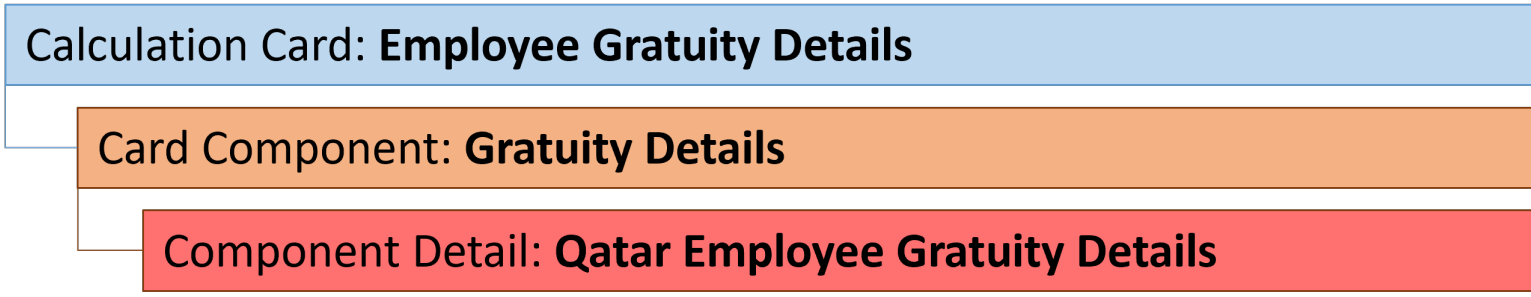
The Employee Gratuity Details calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Employee Gratuity Details calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Employee Gratuity Details'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's termination date.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Employee Gratuity Details Card Component

The Employee Gratuity Details card component captures social insurance related information. It is created for you if the Employee Gratuity Details card is autogenerated.

If you need to update the defaulted gratuity information, provide an Employee Gratuity Details card component and Component Detail record for the Qatar Employee Gratuity Details flexfield context.



Card Component Attributes for Gratuity Details

The Gratuity Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Gratuity Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Gratuity Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Gratuity Details card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Gratuity Details calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Gratuity Details'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.

Component Detail Attributes for Employee Gratuity Details

Employee Gratuity Details uses one flexfield context. Use the Component Detail record type to load the flexfield segment values. In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts: - Qatar Employee Gratuity Details (ORA_HRX_QA_GRATUITY_PREL)

> **Note:** You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Gratuity Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Gratuity Details card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context. Supply ORA_HRX_QA_GRATUITY_PREL.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Employee Gratuity Details for Qatar
- Guidelines for Associating Employee Gratuity Details Cards for Qatar
- Guidelines for Loading Flexfield Data

Example of Updating Employee Gratuity Details for Qatar

This example updates an auto-generated Gratuity Details card component of the Employee Gratuity Details card.

The Qatar Employee Gratuity Details flexfield context is populated with these values for employee assignment E1591:

Flexfield Segment	Value
Override Amount	10050.00
Latest Gratuity Payment Date	31 Dec 2022

The CalculationCard.dat file is used to update Employee Gratuity Details calculation cards with HCM Data Loader.

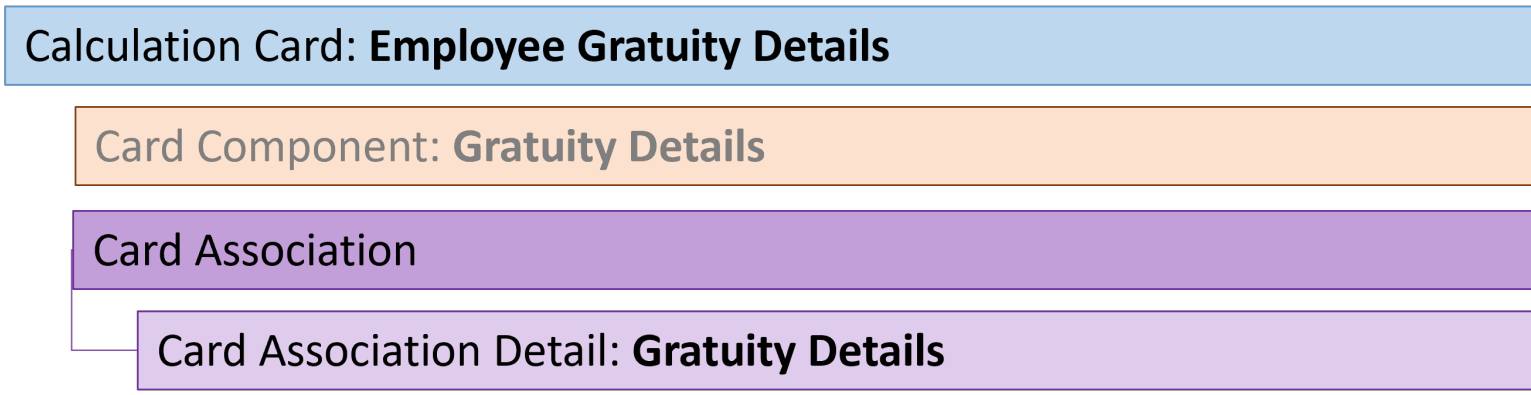
```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber
MERGE|CalculationCard|QA LDG|Employee Gratuity Details|1|2022/12/15|E1591

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|QA LDG|Employee Gratuity Details|1|2022/12/15|E1591|Gratuity Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
oraHrxQaOverrideAmount(Deduction Developer DF=ORA_HRX_QA_GRATUITY_PREL)|oraHrxQaLstGratPymtDate(Deduction
 Developer DF=ORA_HRX_QA_GRATUITY_PREL)
MERGE|ComponentDetail|QA LDG|Employee Gratuity Details|1|2022/12/15|E1591|Gratuity Details|1|
ORA_HRX_QA_GRATUITY_PREL|ORA_HRX_QA_GRATUITY_PREL|10050.00|2022/12/31
```

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Employee Gratuity Details for Qatar
- Guidelines for Loading Flexfield Data

Guidelines for Associating Employee Gratuity Details Cards for Qatar

The card association record associates the Employee Gratuity Details calculation card with the employee's tax reporting unit.



The associated tax reporting unit is defined in the card association. The card association detail allows the **Gratuity Details** card component to be associated with the employee's assignments.

Card Association Attributes for Employee Gratuity Details

The Card Association record type associates the Employee Gratuity Details card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Employee Gratuity Details card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Gratuity Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Tax Withholding calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Employee Gratuity Details calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Employee Gratuity Details

The card association detail associates the Gratuity Details card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Thecard component this association is for. Supply either the source system ID value or user key attributes to identify the Gratuity Details card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of Employee Gratuity Details for Qatar
- Guidelines for Loading Employee Gratuity Details for Qatar

Example of Loading Employee Gratuity Details Card Associations for Qatar

This example creates the tax reporting unit association for the existing Employee Gratuity Details card belonging to employee assignment E1591.

User keys are provided to reference the existing Employee Gratuity Details card and Employee Gratuity Details card component on the assumption they were auto generated and therefore the source key values are unknown.

The CalculationCard.dat file is used to upload the Employee Gratuity Details card associations with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|QA LDG|Employee Gratuity Details|2022/12/15|E1591|1
```

```
METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|QA LDG|Employee Gratuity Details|2022/12/15|E1591|1|Gratuity Details|1

METADATA|CardAssociation|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|TaxReportingUnitName
MERGE|CardAssociation|QA LDG|Employee Gratuity Details|2022/12/15|E1591|1|QA Legal Entity

METADATA|CardAssociationDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|
AssignmentNumber|CardSequence|DirCardCompDefName|ComponentSequence|AssociationAssignmentNumber|
TaxReportingUnitName
MERGE|CardAssociationDetail|QA LDG|Employee Gratuity Details|2022/12/15|E1591|1|Gratuity Details|1|E1591|QA
 Legal Entity
```

When associating an auto-generated card you don't need to include the CardComponent unless also making changes to its values.

Example of Creating Employee Gratuity Details for Qatar

This example creates the Employee Gratuity Details calculation card with associations for employee assignment E3434 on the 1st August 2022, for a payment date of 31st December 2022.

If your Employee Gratuity Details cards are not auto-generated it is recommended that you use source keys to create the card.

The Qatar Employee Gratuity Details flexfield context is populated with these values

Flexfield Segment	Value
Gratuity Override Amount	10000.00
Latest Gratuity Payment Date	31 Dec 2022

The CalculationCard.dat file is used to upload Employee Gratuity Details calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|EGC-E3434|QA LDG|Employee Gratuity Details|2022/08/01|E3434

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|GC-E3434|QA LDG|EGC-E3434|2022/08/01|Gratuity Details

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|oraHrxQaOverrideAmount(Deduction Developer DF=ORA_HRX_QA_GRATUITY_PREL)|
oraHrxQaLstGratPymtDate(Deduction Developer DF=ORA_HRX_QA_GRATUITY_PREL)
MERGE|ComponentDetail|VISION|EGC-E3434|QA LDG|GC-E3434|2022/08/01|Gratuity Details|ORA_HRX_QA_GRATUITY_PREL|
ORA_HRX_QA_GRATUITY_PREL|10000.00|2022/08/31

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|EGC_ASS-E3434|QA LDG|EGC-E3434|2022/08/01|QA Legal Entity

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirRepCardId(SourceSystemId)|EffectiveStartDate|DirCardCompId(SourceSystemId)|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|EGC_ADT-E3434|QA LDG|EGC_ASS-E3434|2022/08/01|GC-E3434|E3434
```

Employee Social Insurance

Overview of Employee Social Insurance Details for Qatar

The Employee Social Insurance Details card stores all information required to accurately compute social insurance contributions, such as employee details, annuity information and reference salary.

One Employee Social Insurance Details card must be created for every Qatar employee and Tax Reporting Unit (TRU) that the employee belongs to.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, Employee Social Insurance Details cards are automatically created when a new employee is entered using the New Hire task with a set of values specified at the person level, such as citizenship and social insurance information. The Social Insurance Details card component is also auto-generated. You only populate the component details if you need to supply override values. There may be cases where this information needs to be loaded in bulk:

During data migration:

- Employee Social Insurance information must be uploaded into Oracle Fusion Payroll, to ensure that contributions are calculated correctly. If HCM Data Loader is used to migrate employee records, a default Employee Social Insurance card is automatically created. In most cases, the default won't reflect the employee's actual Social Insurance information, and therefore the card must be updated.

Ongoing bulk updates:

- When you bulk load New Hire information, a default Employee Social Insurance card may be automatically generated (if the new hire records are created through HCM Data Loader or the interface with Taleo). In this case, you need to update the default card with the correct social insurance information.

It is recommended to have a good understanding of the Employee Social Insurance card and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS (Qatar): Payroll Implementation and Functional Considerations (Document ID 2329792.1).

Employee Social Insurance Calculation Card Record Types

The Employee Social Insurance card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

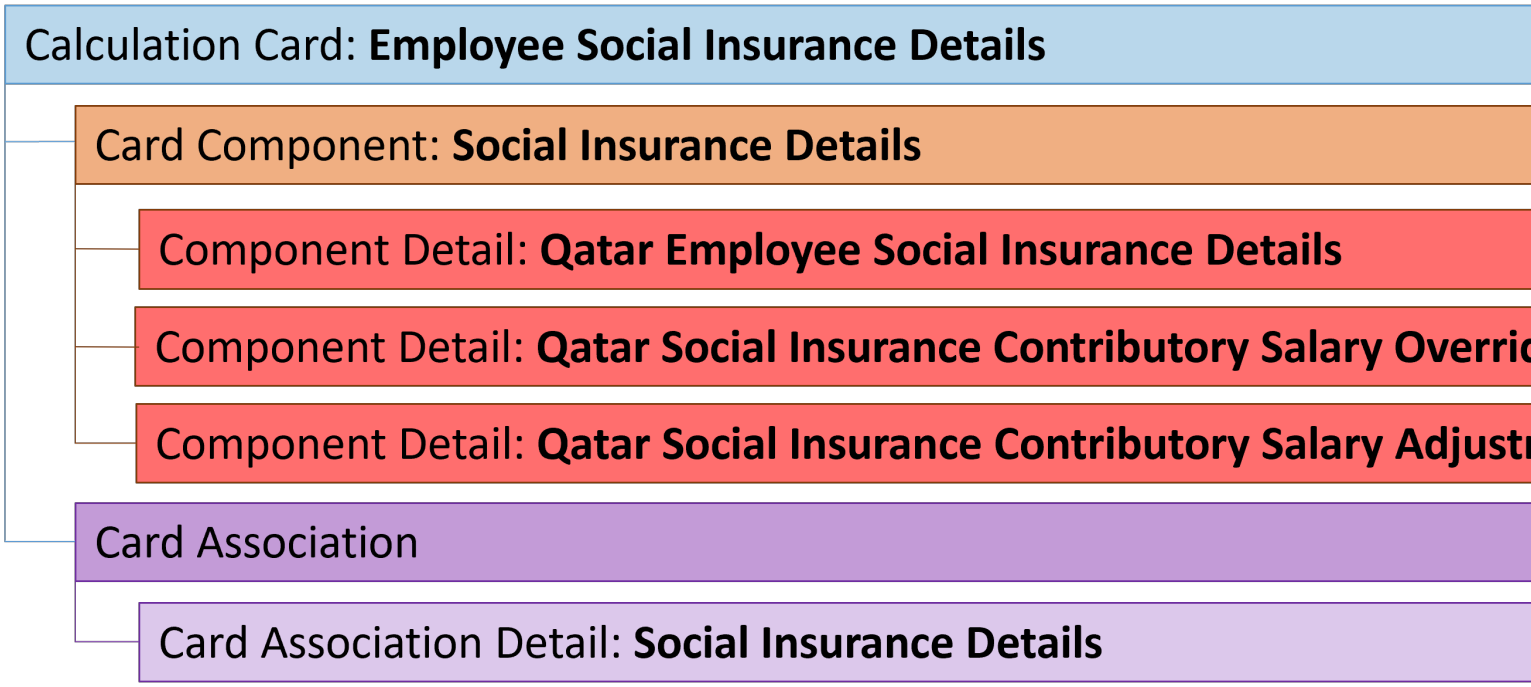
The Employee Social Insurance utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard

Component	Functional Description	File Discriminator
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card comonoent.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Employee Social Insurance Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Employee Social Insurance are described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Employee Social Insurance Details for Qatar
- Guidelines for Associating Employee Social Insurance Details Cards for Qatar
- Examples of Updating Employee Social Insurances Details for Qatar
- Guidelines for Loading Flexfield Data

Guidelines for Loading Employee Social Insurance Details for Qatar

You must have one Calculation Card record for every Qatar employee and TRU they report to.

Even if you are updating an existing Employee Social Insurance Details card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

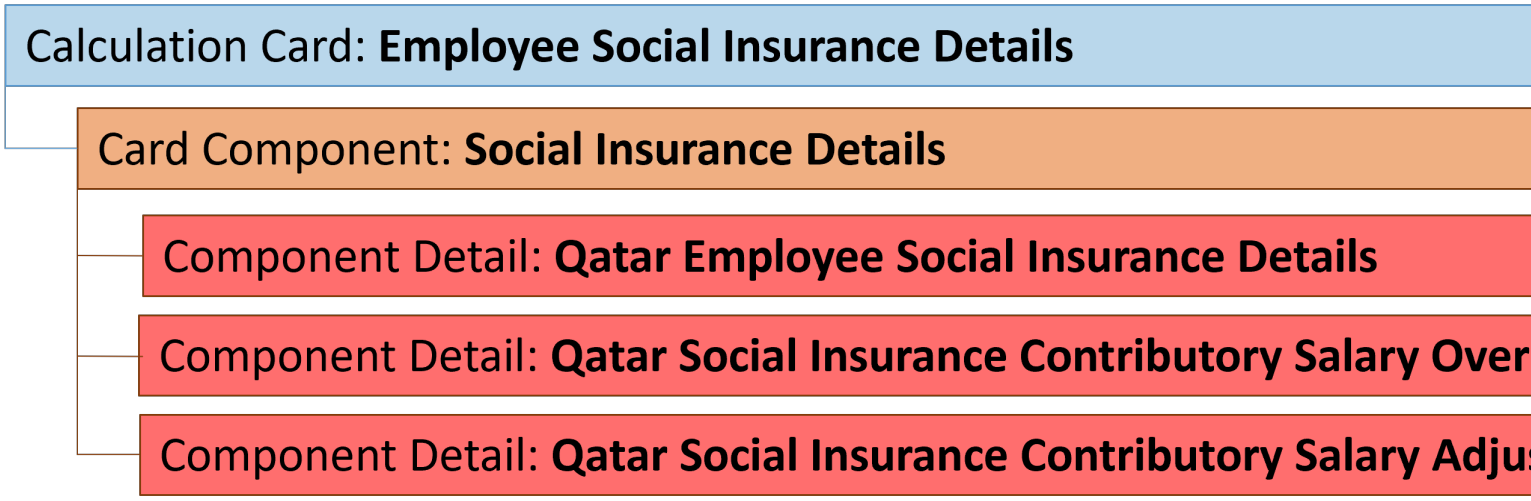
Employee Social Insurance Details Calculation Card Attributes

The Employee Social Insurance Details calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Employee Social Insurance Details calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Employee Social Insurance Details'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Employee Social Insurance Details Card Component

The Social Insurance Details card component captures social insurance related information. It is created for you if the Employee Social Insurance Details card is autogenerated. If you need to update the defaulted social insurance information, provide an Social Insurance Details card component and related Component Detail.



Card Component Attributes for Social Insurance Details Information

The Social Insurance Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Social Insurance Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Social Insurance Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Social Insurance Details card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Social Insurance Details calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Social Insurance Details'.
ComponentSequence	N/A	Specify '1'

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		Note: A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.

Component Detail Attributes for Social Insurance Details

Social Insurance Details uses three flexfield contexts. Use the Component Detail record type to load the flexfield segment values. In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Qatar Employee Social Insurance Details (ORA_HRX_QA_SI_PREL)
- Qatar Social Insurance Contributory Salary Override (ORA_HRX_QA_SI_CONT_SAL_ADJ_PREL)
- Qatar Social Insurance Contributory Salary Adjustment (ORA_HRX_QA_SI_CONT_SAL_ADJ_PREL)

Only the Qatar Employee Social Insurance Details context is required. The others are optional.

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent **Social Insurance Details** card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the **Social Insurance Details** card component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as '**ORA_HRX_QA_SI_PREL**'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Employee Social Insurance Details for Qatar
- Examples of Updating Employee Social Insurances Details for Qatar
- Guidelines for Loading Flexfield Data

Examples of Updating Employee Social Insurances Details for Qatar

These examples update or maintain an auto-generated Employee Social Insurance Details card.

User keys are used to identify the card and card component as the source keys may not be known for auto-generated cards.

Use the CalculationCard.dat file to update Employee Social Insurance Details card component information with HCM Data Loader.

Example of Updating the Auto-Generated Social Insurance Details Card Component

This example updates the existing Employee Social Insurance Details card component for an auto-generated Employee Social Insurace Details card.

The Qatar Employee Social Insurance Details flexfield context is updated, setting these values

Flexfield Segment	Value
Citizenship	Qatar
Registered for Social Insurance	Yes
Exempt from SI Contribution Difference	Yes

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|QA LDG|Employee Social Insurance Details|2022/01/01|E3434|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|QA LDG|Employee Social Insurance Details|2022/01/01|E3434|1|Social Insurance Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
oraHrxQaCitizenship(Deduction Developer DF=ORA_HRX_QA_SI_PREL)|oraHrxQaRegisteredForSi(Deduction Developer
 DF=ORA_HRX_QA_SI_PREL)|oraHrxQaCtzenContryToPay(Deduction Developer DF=ORA_HRX_QA_SI_PREL)
MERGE|ComponentDetail|QA LDG|Employee Social Insurance Details|2022/01/01|E3434|1|Social Insurance Details|
1|ORA_HRX_QA_SI_PREL|ORA_HRX_QA_SI_PREL|QA|Y|Y
```

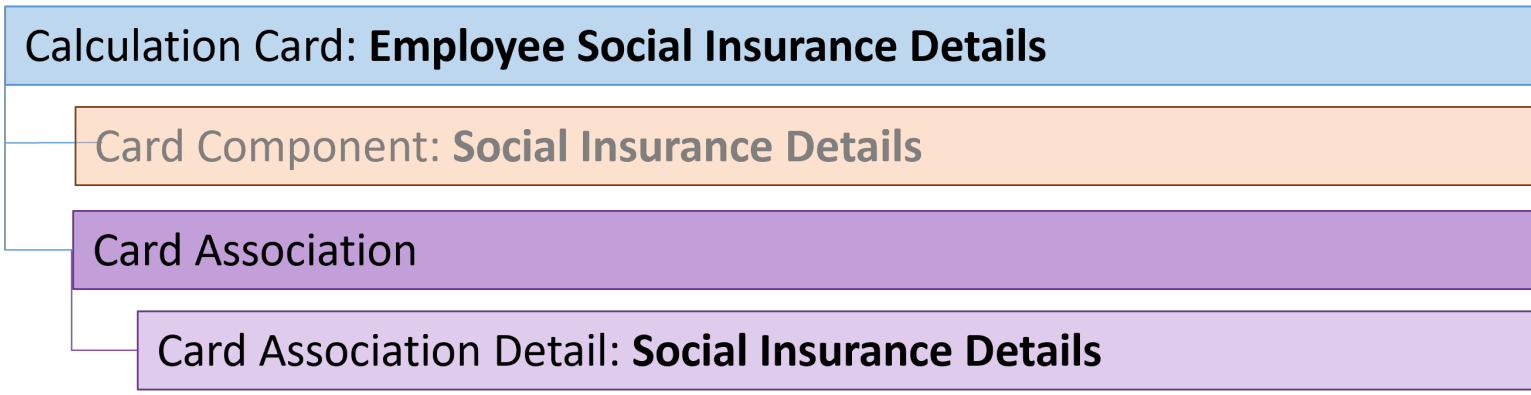
Example of Qatar Social Insurance Contributory Salary Adjustment

This example populates the Qatar Social Insurance Contributory Salary Adjustment flexfield context for an auto-generated Employee Social Insurance Details card. In this example the Adjustment Amount is being set to 5000.00.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|QA LDG|Employee Social Insurance Details|2022/01/01|E3434|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|QA LDG|Employee Social Insurance Details|2022/01/01|E3434|1|Social Insurance Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
oraHrxQaContribSalAdjsmnt(Deduction Developer DF=ORA_HRX_QA_SI_CONT_SAL_ADJ_PREL)
MERGE|ComponentDetail|QA LDG|Employee Social Insurance Details|2022/01/01|E3434|1|Social Insurance Details|
1|ORA_HRX_QA_SI_CONT_SAL_ADJ_PREL|ORA_HRX_QA_SI_CONT_SAL_ADJ_PREL|5000.00
```

Guidelines for Associating Employee Social Insurance Details Cards for Qatar

The card association record associates the Employee Social Insurance Details calculation card with the employee's tax reporting unit.



The associated tax reporting unit is defined in the card association. The card association detail allows the Social Insurance Details card component to be associated with the employee's assignments.

Card Association Attributes for Employee Social Insurance Details

The Card Association record type associates the Employee Social Insurance Details card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Employee Social Insurance Details card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Social Insurance Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Social Insurance Details calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Employee Social Insurace Details calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Social Insurance Details

The card association detail associates the Social Insurance Details card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Thecard component this association is for. Supply either the source system ID value or user key attributes to identify the **Social InsuranceDetails** card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of Employee Social Insurance Details for Qatar
- Guidelines for Loading Employee Social Insurance Details for Qatar

Example of Loading Employee Social Insurance Details Card Associations for Qatar

This example creates the tax reporting unit association for the existing Employee Social Insurance Details card belonging to employee assignment E3434.

User keys are provided to reference the existing Employee Social Insurance Details card and Social Insurance Details card component on the assumption they were auto generated and therefore the source key values are unknown.

The CalculationCard.dat file is used to upload the Employee Social Insurance Details card associations with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
```

```
MERGE|CalculationCard|QA LDG|Employee Social Insurance Details|2022/01/01|E3434|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|QA LDG|Employee Social Insurance Details|2022/01/01|E3434|1|Social Insurance Details|1

METADATA|CardAssociation|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|TaxReportingUnitName
MERGE|CardAssociation|QA LDG|Employee Social Insurance Details|2022/01/01|E3434|1|QA Legal Entity

METADATA|CardAssociationDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|
AssignmentNumber|CardSequence|DirCardCompDefName|ComponentSequence|AssociationAssignmentNumber|
TaxReportingUnitName
MERGE|CardAssociationDetail|QA LDG|Employee Social Insurance Details|2022/01/01|E3434|1|Social Insurance
 Details|1|E3434|QA Legal Entity
```

When associating an auto-generated card, you don't need to include the CardComponent unless also making changes to its values.

Example of Creating Employee Social Insurance Details for Qatar

This example creates an Employee Social Insurance Details card with associations for employee assignment E3434.

If your Employee Social Insurance Details cards are not auto-generated it is recommended that you use source keys to create the card.

The Qatar Employee Social Insurance Details flexfield context is populated with these values

Flexfield Segment	Value
Citizenship	Qatar
Registered for Social Insurance	Yes
Exempt from SI Contribution Difference	Yes

The CalculationCard.dat file is used to upload Employee Social Insurance Details calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|E3434_SI|QA LDG|Employee Social Insurance Details|2022/01/01|E3434

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|E3434_SI_DET|QA LDG|E3434_SI|2022/01/01|Social Insurance Details

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|oraHrxQaCitizenship(Deduction Developer DF=ORA_HRX_QA_SI_PREL)|
oraHrxQaRegisteredForSi(Deduction Developer DF=ORA_HRX_QA_SI_PREL)|oraHrxQaCtzenContryToPay(Deduction
 Developer DF=ORA_HRX_QA_SI_PREL)
MERGE|ComponentDetail|VISION|E3434_SI_PREL|QA LDG|E3434_SI_DET|2022/01/01|Social Insurance Details|
ORA_HRX_QA_SI_PREL|ORA_HRX_QA_SI_PREL|QA|Y|Y

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|E3434_SI_TRU|QA LDG|E3434_SI|2022/01/01|QA Legal Entity
```

```
METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirRepCardId(SourceSystemId)|
DirCardCompId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|E3434_SI_TRU|E3434_SI_TRU|E3434_SI_DET|2022/01/01|E3434
```

Court Orders

Overview of Court Orders for Qatar

The Court Orders card stores information for court order deductions, such as amount, frequency, and payee.

Considerations and Prerequisites

Use the Elements task to create Court Order elements. When you create court order elements, a card component definition is generated with the same name as the element. The generated card component definition is then available to use on the predefined Court Orders calculation card.

Unlike Employee Social Insurance Details or Employee Gratuity Details cards, the Court Order card is not generated automatically. You only need to create this card for those employees who are subject to court order deductions.

There might be cases when there is a need to bulk load Court Orders cards:

During data migration:

Employee's Court Orders must be migrated to the Oracle Payroll Cloud to ensure that appropriate deductions are considered. Using HCM Data Loader you must create the card and all necessary card components and component details.

Ongoing bulk updates:

- New hires can have Court Orders that need to be uploaded in bulk.

It is recommended to have a good understanding of the Court Orders card and the information it contains before attempting mass upload. For further information, see Oracle Fusion HRMS (Qatar): Payroll Implementation and Functional Considerations (Document ID 2329792.1).

Before bulk loading Court Order information:

- Create eligibility for the Court Order element.
- Create third-party payee and third-party payment methods for the people and organizations the payment of the court order is made to.
- Update the PAY_INVLN_DEDN_ORDER_AMOUNT_PAYEE lookup type with the court order issuing authorities.

Court Order Calculation Card Record Types

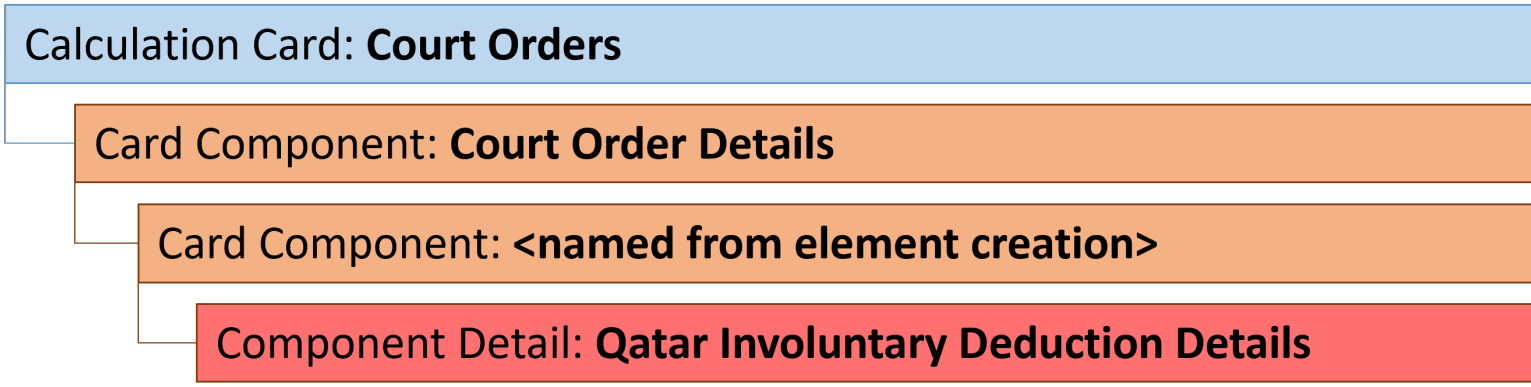
The Court Orders card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Court Orders card uses the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail

Court Orders Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Court Orders is described in this diagram:



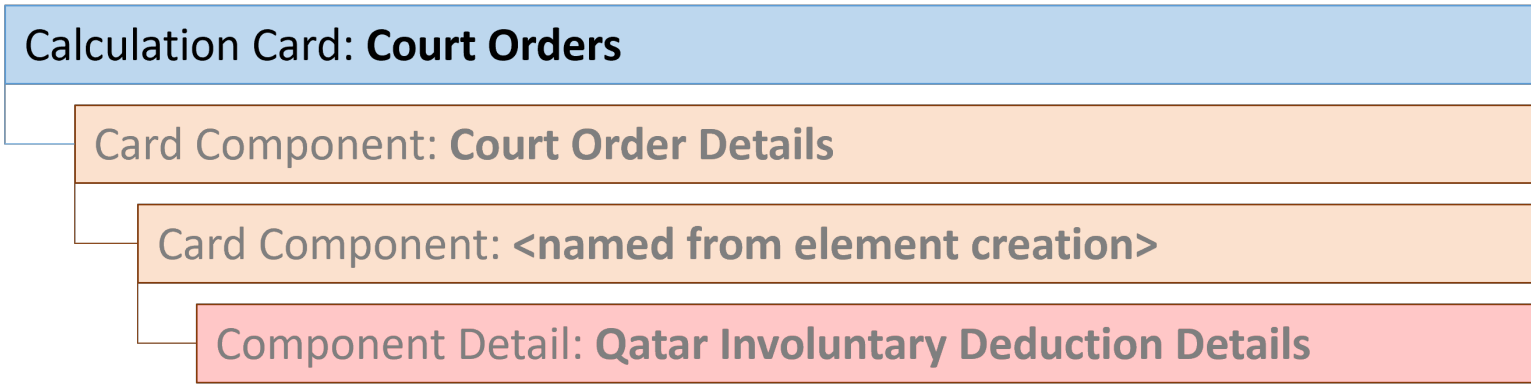
Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Court Orders for Qatar
- Guidelines for Loading Flexfield Data

Guidelines for Loading Court Orders for Qatar

The Court Orders calculation card record needs to be included even if you're updating an existing Court Orders card.

Even if the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.



The Court Orders calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Court Orders calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Court Orders'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card.
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Court Order Details Card Component

The Court Order Details card component is a predefined card component, which acts as a parent to the element-specific card component.

Calculation Card: **Court Orders**

Card Component: **Court Order Details**

Card Component: **<named from element creation>**

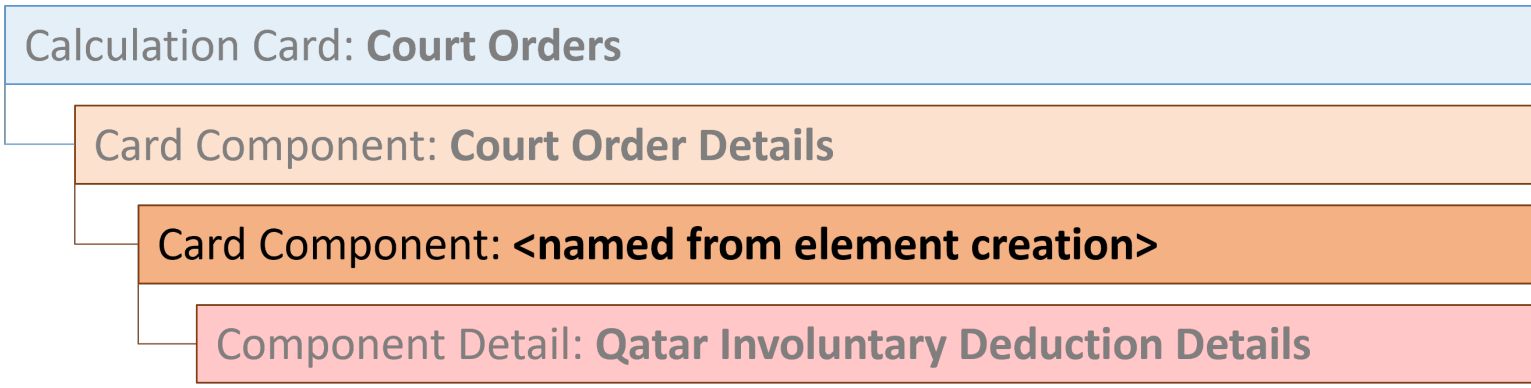
Component Detail: **Qatar Involuntary Deduction Details**

The Court Order Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the **Court Orders Details** card component. For new card components,, supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Court Order Details card component. This must be the same or after the EffectiveStartDate on the Court Orders calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Court Orders Details'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.

Card Component Named for the Court Order Element

This card component is a named for the court orders element and is created as a child to the Court Order Details card component.

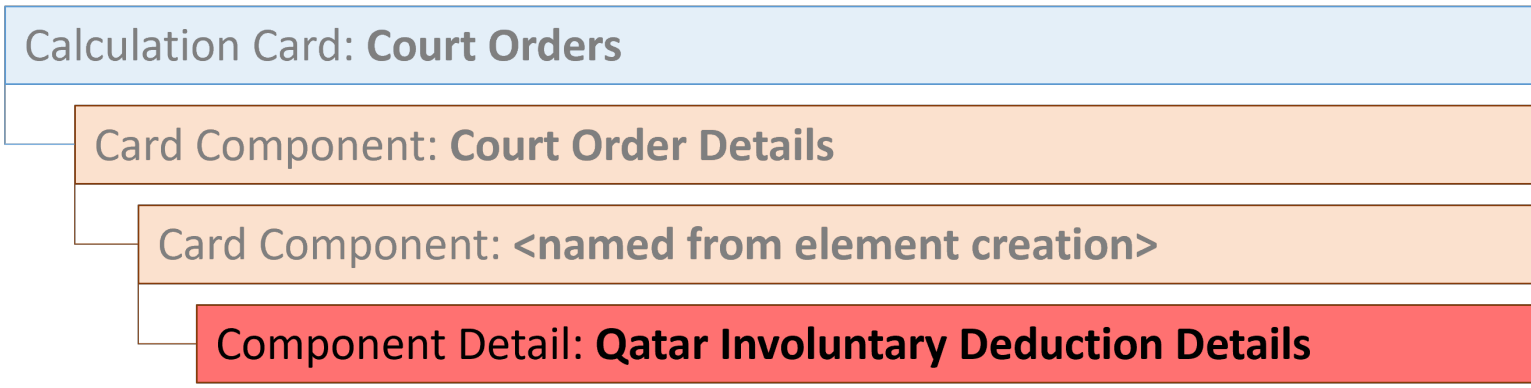


This card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the card component. For new card components,card ,card supply the source key attributes. You can also identify components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
ParentDirCardCompId (SourceSystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	The parent Court Order Details card component. Use the same key type used to identify the Court Order Details card component.
EffectiveStartDate	N/A	The start date of the card component. This must be the same as the EffectiveStartDate on the Court Orders calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify the name defined when creating the court order element.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.
Context1	N/A	Reference code of the court order.

Component Detail Attributes for Court Orders



Court Orders uses a single flexfield context. Use the Component Detail record type to load the flexfield segment values. In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Qatar Involuntary Deduction Details (ORA_HRX_QA_INVOL_DED_DETAILS)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

To supply third-party details on the Qatar Involuntary Deduction Details flexfield context, refer to the Cloud Customer Connect topic ***BI Publisher Report: Third Party Payees for Involuntary Deduction Cards*** for a report that extracts the third-party names and party IDs for the country, legislation and payee type specified.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName,	The parent card component should be referenced using the same key type used to

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, DirCardCompDefName	identify the parent record. This is the card component that is named after the Court Order element. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parentcard component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context. Supply ORA_HRX_SA_INVOL_DED_DETAILS. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Court Orders for Qatar
- Example of Creating Court Orders for Qatar
- Guidelines for Loading Flexfield Data

Example of Creating Court Orders for Qatar

This example creates a Court Orders card for employee assignment E3434.

The Qatar Involuntary Deduction Details flexfield context is populated with these values

Flexfield Segment	Value
Amount	1000
Date of Issue	7 July 2022

Flexfield Segment	Value
Order Amount Payee Details	QA Court Order Payee

The CalculationCard.dat file is used to upload Court Orders calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
AssignmentNumber|EffectiveStartDate
MERGE|CalculationCard|VISION|E3434_CO|QA LDG|Court Orders|E3434|2022/07/07

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
DirCardCompDefName|EffectiveStartDate|Context1|ParentDirCardCompId(SourceSystemId)
MERGE|CardComponent|VISION|E3434_COD|QA LDG|E3434_CO|Court Order Details|2022/07/07||
MERGE|CardComponent|VISION|E3434_QACO|QA LDG|E3434_CO|QA Court Order|2022/07/07|Court Order Reference|
E3434_COD

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|DirCardCompDefName|EffectiveStartDate|DirInformationCategory|
FLEX:Deduction Developer DF|oraHrxQaAmount(Deduction Developer DF=ORA_HRX_QA_INVOL_DED_DETAILS)|
oraHrxQaDateOfIssue(Deduction Developer DF=ORA_HRX_QA_INVOL_DED_DETAILS)|
oraHrxQaThirdPartyPayee_Display(Deduction Developer DF=ORA_HRX_QA_INVOL_DED_DETAILS)
MERGE|ComponentDetail|VISION|E3434_QACO_ID_DET|QA LDG|E3434_QACO|QA Court Order|2022/07/07|
ORA_HRX_QA_INVOL_DED_DETAILS|ORA_HRX_QA_INVOL_DED_DETAILS|1000.00|2022/07/07|QA Court Order Payee
```

31 Loading Payroll Localization Data for Saudi Arabia

Employee GOSI Details

Overview of Employee GOSI Details for Saudi Arabia

The Employee GOSI Details card stores all information required to accurately compute social insurance contributions, such as employee details, annuity information, and reference salary.

One Employee GOSI Details card must be created for every Saudi Arabia employee and Tax Reporting Unit (TRU) that the employee belongs to.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, Employee GOSI Details cards are automatically created when a new employee is entered using the New Hire task with a set of values specified at the person level, such as citizenship and GOSI number. Only the Employee GOSI Details card component is auto-generated. You populate the component details if you need to supply override values. There can be cases where this information needs to be loaded in bulk:

During data migration:

Employee GOSI Details information must be uploaded into Oracle Fusion Payroll, to ensure that contributions are calculated correctly. If HCM Data Loader is used to migrate employee records, a default Employee GOSI Details card is automatically created. In most cases, the default won't reflect the employee's actual GOSI information, and therefore the card must be updated.

Ongoing bulk updates:

- When you bulk load New Hire information, a default Employee GOSI Details card can be automatically generated (if the new hire records are created through HCM Data Loader or the interface with Taleo). In this case, you need to update the default card with the correct GOSI information.

It is recommended to have a good understanding of the Employee GOSI Details card and the information it contains before attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS (Saudi Arabia): Payroll Implementation and Functional Considerations (Document ID 1619159.1).

Employee GOSI Details Calculation Card Record Types

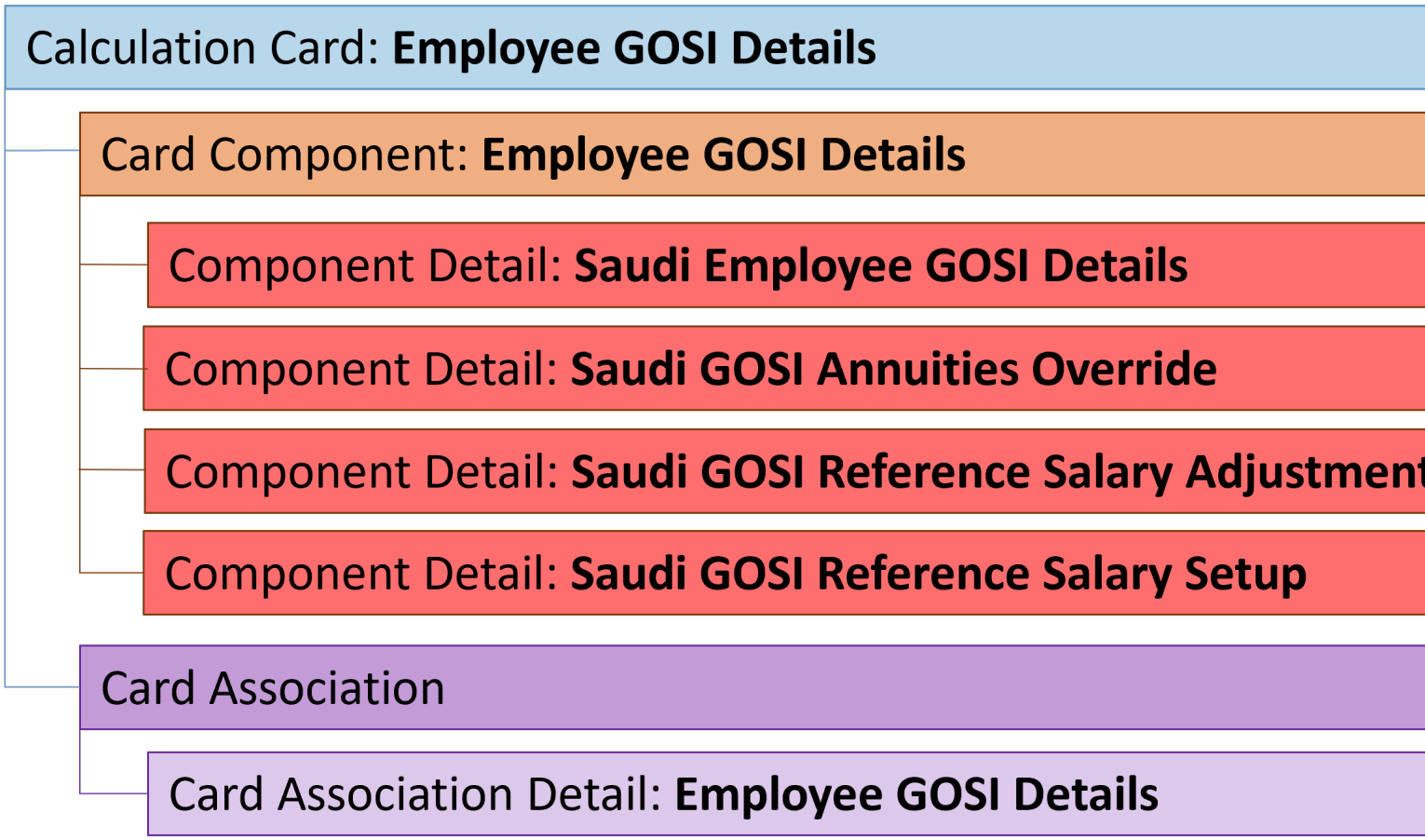
The Employee GOSI Details card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Employee GOSI Details uses these record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Employee GOSI Details Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Employee GOSI Details is described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Employee GOSI Details for Saudi Arabia
- Guidelines for Associating Employee GOSI Details Cards for Saudi Arabia
- Examples of Updating Employee GOSI Details for Saudi Arabia
- Guidelines for Loading Flexfield Data

Guidelines for Loading Employee GOSI Details for Saudi Arabia

You must have one Calculation Card record for every Saudi Arabia employee you are maintaining GOSI Details for.

Even if you are updating an existing Employee GOSI Details card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Employee GOSI Details Calculation Card Attributes

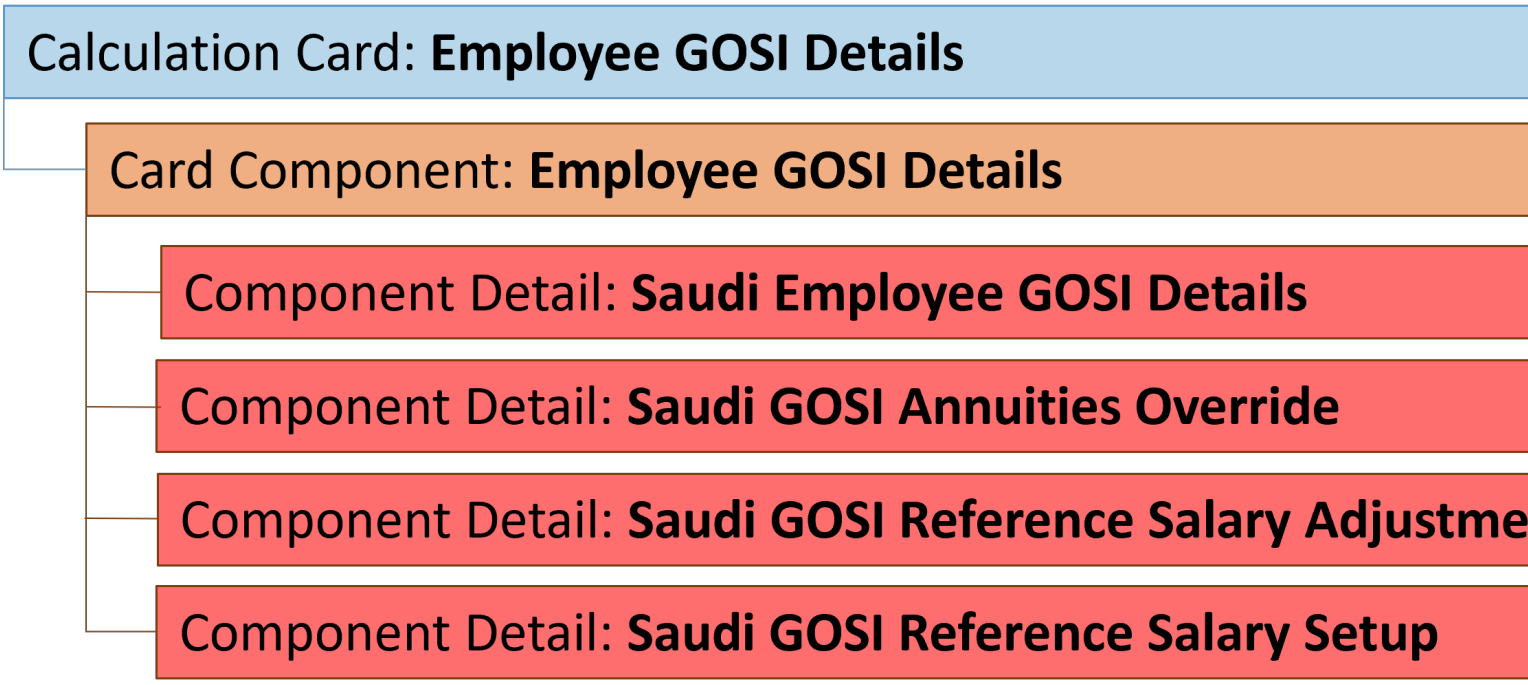
The Employee GOSI Details calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Employee GOSI Details calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Employee GOSI Details'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Employee GOSI Details Card Component

The Employee GOSI Details card component captures social insurance-related information. It is created for you if the Employee GOSI Details card is autogenerated.

If you need to update the defaulted GOSI information, provide an Employee GOSI Details card component and related Component Detail.



Card Component Attributes for Employee GOSI Details Information

The Employee GOSI Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee GOSI Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee GOSI Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Employee GOSI Details card component, typically the employee's start date. This must be the same as the

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		EffectiveStartDate on the Employee GOSI Details calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Employee GOSI Details'.
ComponentSequence	N/A	Specify '1'.

Component Detail Attributes for Employee GOSI Details

Employee GOSI Details uses four flexfield contexts. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Saudi Employee GOSI Details (HRX_SA_GOSI_PREL)
- Saudi GOSI Annuities Override (HRX_SA_GOSI_ANNUITIES_OVERRIDE_PREL)
- Saudi GOSI Reference Salary Adjustment (HRX_SA_GOSI_ADJUST_PREL)
- Saudi GOSI Reference Salary Setup (HRX_SA_GOSI_SETUP_PREL)

Only the **Saudi Employee GOSI Details** context is required. The others are optional.

> **Note:** You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Employee GOSI Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee GOSI Details card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_SA_GOSI_PREL'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Employee GOSI Details for Saudi Arabia
- Examples of Updating Employee GOSI Details for Saudi Arabia
- Guidelines for Loading Flexfield Data

Examples of Updating Employee GOSI Details for Saudi Arabia

These examples update or maintain an auto-generated Employee GOSI Details card. User keys are used to identify the card and card component as the source keys might not be known for auto-generated cards.

Use the CalculationCard.dat file to update Employee GOSI Details card component information with HCM Data Loader.

Example of Updating the Auto-Generated Employee GOSI Details Card Component

This example updates the existing Employee GOSI Details card component for an auto-generated Employee GOSI Details card.

The Saudi Employee GOSI Details flexfield context is updated, setting these values

Flexfield Segment	Value
Citizenship	Saudi Arabia
Registered for Social Insurance	Yes
Annuities	Yes

Flexfield Segment	Value
Hazards	Yes

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|SA LDG|Employee GOSI Details|2019/01/01|E195516|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|SA LDG|Employee GOSI Details|2019/01/01|E195516|1|Employee GOSI Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardCompDefName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer
 DF|_CITIZENSHIP(Deduction Developer DF=HRX_SA_GOSI_PREL)|_REGISTERED_FOR_SI(Deduction Developer
 DF=HRX_SA_GOSI_PREL)|_ANNUITIES(Deduction Developer DF=HRX_SA_GOSI_PREL)|_ANNUITIES_AGE_60_65(Deduction
 Developer DF=HRX_SA_GOSI_PREL)|_HAZARDS(Deduction Developer DF=HRX_SA_GOSI_PREL)
MERGE|ComponentDetail|SA LDG|Employee GOSI Details|2019/01/01|E195516|1|Employee GOSI Details|1|
HRX_SA_GOSI_PREL|HRX_SA_GOSI_PREL|SA|Y|Y||Y
```

Example of Creating Setup GOSI Reference Salary

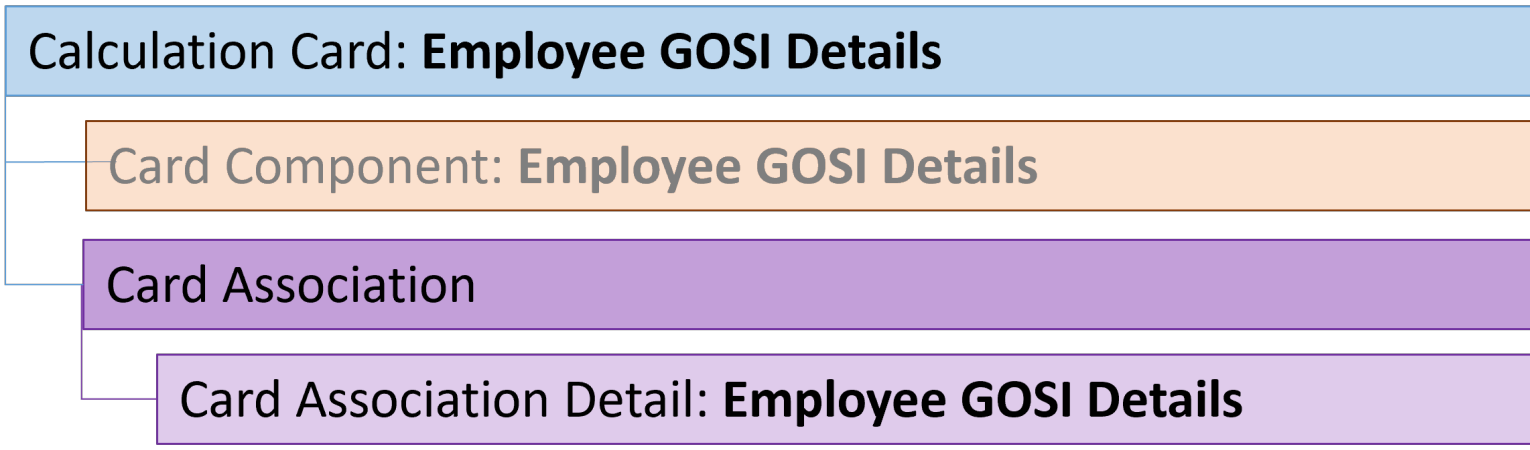
This example populates the Saudi GOSI Reference Salary Setup flexfield context for an auto-generated Employee GOSI Details card.

In this example the Setup GOSI Reference Salary is being set to 5000.00.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|SA LDG|Employee GOSI Details|2019/01/01|E191705|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|SA LDG|Employee GOSI Details|2019/01/01|E191705|1|Employee GOSI Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
_SETUP_GOSI_REF_SALARY(Deduction Developer DF=HRX_SA_GOSI_SETUP_PREL)
MERGE|ComponentDetail|SA LDG|Employee GOSI Details|2019/01/01|E191705|1|Employee GOSI Details|1|
HRX_SA_GOSI_SETUP_PREL|HRX_SA_GOSI_SETUP_PREL|5000.00
```

Guidelines for Associating Employee GOSI Details Cards for Saudi Arabia

The card association record associates the Employee GOSI Details calculation card with the employee's tax reporting unit.



The associated tax reporting unit is defined in the card association. The card association detail allows the Employee GOSI Details card component to be associated with the employee's assignments.

Card Association Attributes for Employee GOSI Details

The Card Association record type associates the Employee GOSI Details card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Employee GOSI Details card association. For new card associations, supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee GOSI Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee GOSI Details calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Employee GOSI Details calculation card. If source keys are used to identify the

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Employee GOSI Details

The card association detail associates the Employee GOSI Details card component with the payroll assignments for the employee.

If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations, supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Thecard component this association is for. Supply either the source system ID value or user key attributes to identify the Employee GOSI Details card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of Employee GOSI Details for Saudi Arabia
- Guidelines for Loading Employee GOSI Details for Saudi Arabia

Employee Gratuity Details

Overview of Employee Gratuity Details for Saudi Arabia

The Employee Gratuity Details card stores all information required to calculate and process the gratuity payment, such as employee gratuity details and information regarding Article 77.

The Employee Gratuity Details card is created automatically upon employee termination and the gratuity amount calculated. When the gratuity card is created, the Employee Gratuity Details component details and card association are also created.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, Employee Gratuity Details cards are automatically created upon employee termination. There may be cases where this information needs to be loaded in bulk:

During data migration:

- Employee Gratuity Details information must exist into Oracle Fusion Payroll in order to ensure that values are calculated correctly. If HCM Data Loader is used to migrated employee termination records, a default Employee Gratuity Details card is automatically created. For historical reasons you may wish to upload gratuity calculation cards to records the Last Gratuity Payment Date for a previous employment.

Ongoing bulk updates:

- When you bulk load employee terminations a default Employee Gratuity Details calculation card may be generated automatically (if the terminations records are created using HCM Data Loader). In this case, you need to update the default card with the correct gratuity information and, if needed, create a component detail record to store a gratuity override amount

It is recommended to have a good understanding of the **Employee Gratuity Details** card and the information it contains prior to attempting mass upload as it has a direct impact on calculations and reporting. For further information, see Oracle Fusion HRMS (Saudi Arabia): Payroll Implementation and Functional Considerations (Document ID 1619159.1).

Employee Gratuity Details Calculation Card Record Types

The Employee Gratuity Details card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

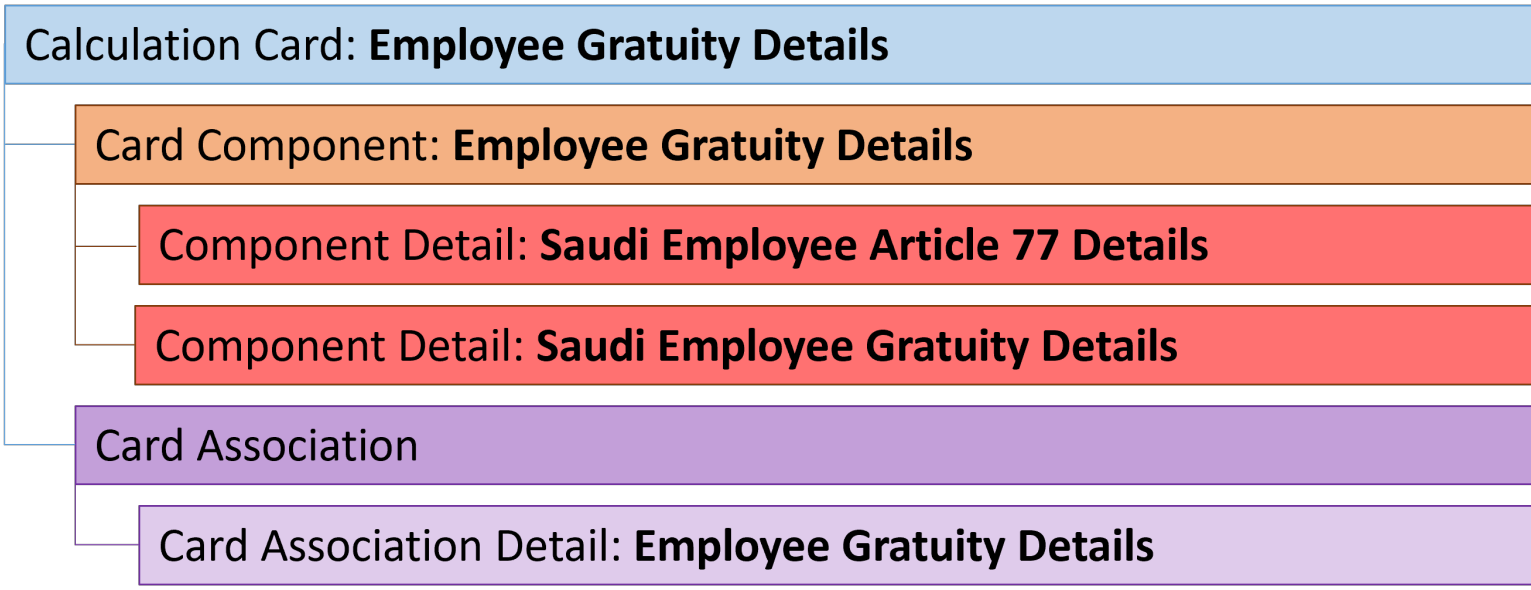
The Employee Gratuity Details utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard

Component	Functional Description	File Discriminator
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Employee Gratuity Details Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Employee Gratuity Details are described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Employee Gratuity Details for Saudi Arabia
- Guidelines for Associating Employee Gratuity Details Cards for Saudi Arabia
- Example of Updating Employee Gratuity Details for Saudi Arabia
- Guidelines for Loading Flexfield Data

Guidelines for Loading Employee Gratuity Details for Saudi Arabia

You must have one Calculation Card record for every Saudi Arabia employee you are maintaining Gratuity Details for.

Even if you are updating an existing Employee Gratuity Details card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Employee Gratuity Details Calculation Card Attributes

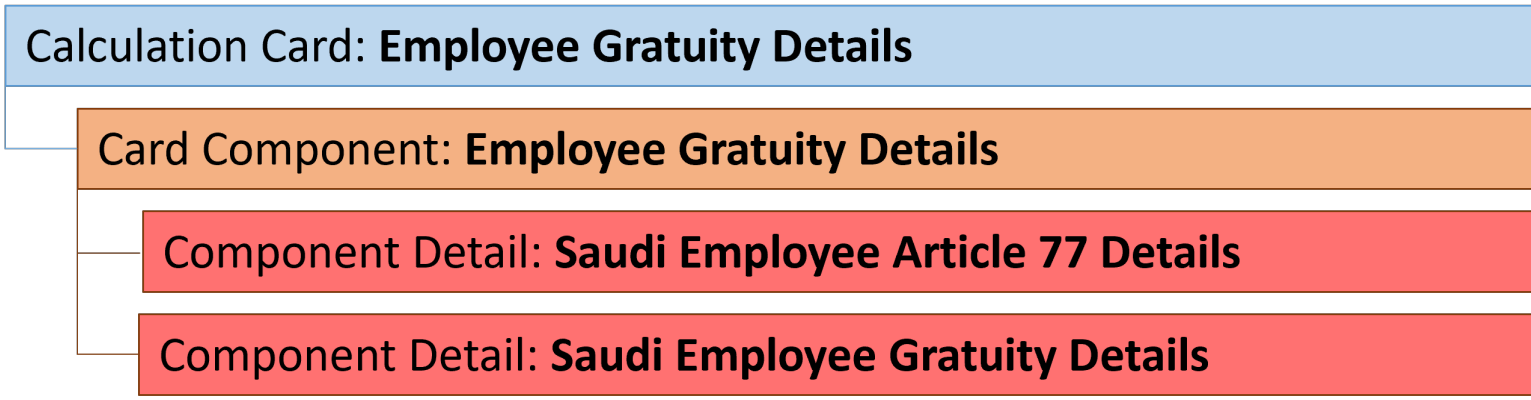
The Employee Gratuity Details calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Employee Gratuity Details calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify '**Employee Gratuity Details**'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's termination date.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Employee Gratuity Details Card Component

The Employee Gratuity Details card component captures social insurance related information. It is created for you if the Employee Gratuity Details card is autogenerated.

If you need to update the defaulted gratuity information, provide an Employee Gratuity Details card component and related Component Detail for each flexfield context relevant to the terminated employee.



Card Component Attributes for Employee Gratuity Details

The Employee Gratuity Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee Gratuity Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Gratuity Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Employee Gratuity Details card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Gratuity Details calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Employee Gratuity Details'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.

Component Detail Attributes for Employee Gratuity Details

Employee Gratuity Details uses two flexfield contexts. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Saudi Employee Article 77 (HRX_SA_ARTICLE77_PREL)
- Saudi Employee Gratuity Details (HRX_SA_GRATUITY_PREL)

Only the Saudi Employee Gratuity Details context is required. The others are optional.

> **Note:** You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Employee Gratuity Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Gratuity Details card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as HRX_SA_GRATUITY_PREL. Flexfield context codes

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
		are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Employee Gratuity Details for Saudi Arabia
- Guidelines for Associating Employee Gratuity Details Cards for Saudi Arabia
- Guidelines for Loading Flexfield Data

Example of Updating Employee Gratuity Details for Saudi Arabia

This example updates an auto-generated Employee Gratuity Details card component of the Employee Gratuity Details card.

The Saudi Employee Gratuity Details flexfield context is populated with these values for employee assignment E191668:

Flexfield Segment	Value
Gratuity Override Amount	1050.00
Latest Gratuity Payment Date	31 Dec 2022

The CalculationCard.dat file is used to update Employee Gratuity Details calculation cards with HCM Data Loader.

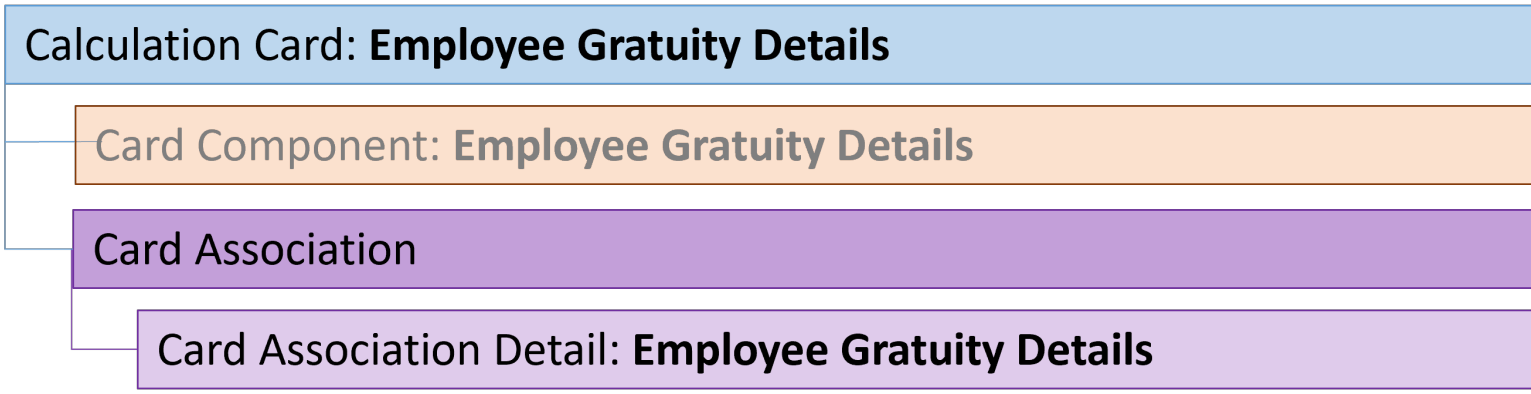
```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber
MERGE|CalculationCard|SA LDG|Employee Gratuity Details|1|2022/07/01|E191668

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence|Context1
MERGE|CardComponent|SA LDG|Employee Gratuity Details|1|2022/07/01|E191668|Employee Gratuity Details|11|11

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
_GRATUITY_OVERRIDE_AMOUNT(Deduction Developer DF=HRX_SA_GRATUITY_PREL)|oraHrxSaLstGratPymtDate(Deduction
 Developer DF=HRX_SA_GRATUITY_PREL)
MERGE|ComponentDetail|SA LDG|Employee Gratuity Details|1|2022/07/01|E191668|Employee Gratuity Details|11|
HRX_SA_GRATUITY_PREL|HRX_SA_GRATUITY_PREL|1050.00|2022/12/31
```

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Employee Gratuity Details for Saudi Arabia
- Guidelines for Loading Flexfield Data

Guidelines for Associating Employee Gratuity Details Cards for Saudi Arabia

The card association record associates the Employee Gratuity Details calculation card with the employee's tax reporting unit.



The associated tax reporting unit is defined in the card association. The card association detail allows the Employee Gratuity Details card component to be associated with the employee's assignments.

Card Association Attributes for Employee Gratuity Details

The Card Association record type associates the Employee Gratuity Details card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Employee Gratuity Details card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Gratuity Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Gratuity Details calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Employee Gratuity Details calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Employee Gratuity Details

The card association detail associates the Employee Gratuity Details card component with the payroll assignments for the employee.

If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Thecard component this association is for. Supply either the source system ID value or user key attributes to identify the Employee Gratuity Details card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of Employee Gratuity Details for Saudi Arabia
- Guidelines for Loading Employee Gratuity Details for Saudi Arabia

Example of Loading Employee Gratuity Details Card Associations for Saudi Arabia

This example creates the tax reporting unit association for the existing Employee Gratuity Details card belonging to employee assignment E191668.

User keys are provided to reference the existing Employee Gratuity Details card and Employee Gratuity Details card component on the assumption they were auto generated and therefore the source key values are unknown.

The CalculationCard.dat file is used to upload the Employee Gratuity Details card associations with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|SA LDG|Employee Gratuity Details|2022/07/01|E191668|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|Context1
MERGE|CardComponent|SA LDG|Employee Gratuity Details|2022/07/01|E191668|1|Employee Gratuity Details|11|11

METADATA|CardAssociation|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|TaxReportingUnitName
MERGE|CardAssociation|SA LDG|Employee Gratuity Details|2022/07/01|E191668|1|SA Legal Entity

METADATA|CardAssociationDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|
AssignmentNumber|CardSequence|DirCardCompDefName|ComponentSequence|AssociationAssignmentNumber|
TaxReportingUnitName
MERGE|CardAssociationDetail|SA LDG|Employee Gratuity Details|2022/07/01|E191668|1|Employee Gratuity Details|
11|E191668|SA Legal Entity
```

When associating an auto-generated card, you don't need to include the CardComponent unless also making changes to its values.

Example of Creating Employee Gratuity Details for Saudi Arabia

This example creates the Employee Gratuity Details calculation card with associations for employee assignment E191668 on the 1st January 2022, for a payment date of 31st December 2022.

If your Employee Gratuity Details cards are not auto-generated it is recommended that you use source keys to create the card.

The Saudi Employee Gratuity Details flexfield context is populated with these values:

Flexfield Segment	Value
Gratuity Override Amount	1000.00

Flexfield Segment	Value
Latest Gratuity Payment Date	31 Dec 2022

The CalculationCard.dat file is used to upload Employee Gratuity Details calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|E191668_GD|SA LDG|Employee Gratuity Details|2022/01/01|E191668

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName|Context1
MERGE|CardComponent|VISION|E191668_GD_11|SA LDG|E191668_GD|2022/01/01|Employee Gratuity Details|11

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|_GRATUITY_OVERRIDE_AMOUNT(Deduction Developer DF=HRX_SA_GRATUITY_PREL)|
oraHrxSaLstGratPymtDate(Deduction Developer DF=HRX_SA_GRATUITY_PREL)
MERGE|ComponentDetail|VISION|E191668_GD_11_GRAT_PREL|SA LDG|E191668_GD_11|2022/01/01|Employee Gratuity
 Details|HRX_SA_GRATUITY_PREL|HRX_SA_GRATUITY_PREL|1000.00|2022/12/31

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|E191668_GD_11_TRU|SA LDG|E191668_GD|2022/01/01|SA Legal Entity

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirRepCardId(SourceSystemId)|EffectiveStartDate|DirCardCompId(SourceSystemId)|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|E191668_GD_11_AD|SA LDG|E191668_GD_11_TRU|2022/01/01|E191668_GD_11|
E191668
```

Court Orders

Overview of Court Orders for Saudi Arabia

The Court Orders card stores information for court order deductions, such as amount, frequency, and payee.

Considerations and Prerequisites

Use the **Elements** task to create Court Order elements. When you create court order elements, a card component definition is generated with the same name as the element. The generated card component definition is then available to use on the predefined Court Orders calculation card.

Unlike Employee GOSI Details or Employee Gratuity Details cards, the Court Order card is not generated automatically. You only need to create this card for those employees who are subject to court order deductions.

There can be cases when there is a need to bulk load **Court Orders** cards:

During data migration:

Employee's Court Orders must be migrated to the Oracle Payroll Cloud to ensure that appropriate deductions are considered. Using HCM Data Loader you must create the card and all necessary card components and component details.

Ongoing bulk updates:

- New hires can have Court Orders that need to be uploaded in bulk.

It is recommended to have a good understanding of the Court Orders card and the information it contains before attempting mass upload. For further information, see Oracle Fusion HRMS (Saudi Arabia): Payroll Implementation and Functional Considerations (Document ID 1619159.1).

Before bulk loading Court Order information:

- Create eligibility for the Court Order element.
- Create third-party payee and third-party payment methods for the people and organizations the payment of the court order is made to.
- Update the PAY_INVLN_DEDN_ORDER_AMOUNT_PAYEE lookup type with the court order issuing authorities.

Court Order Calculation Card Record Types

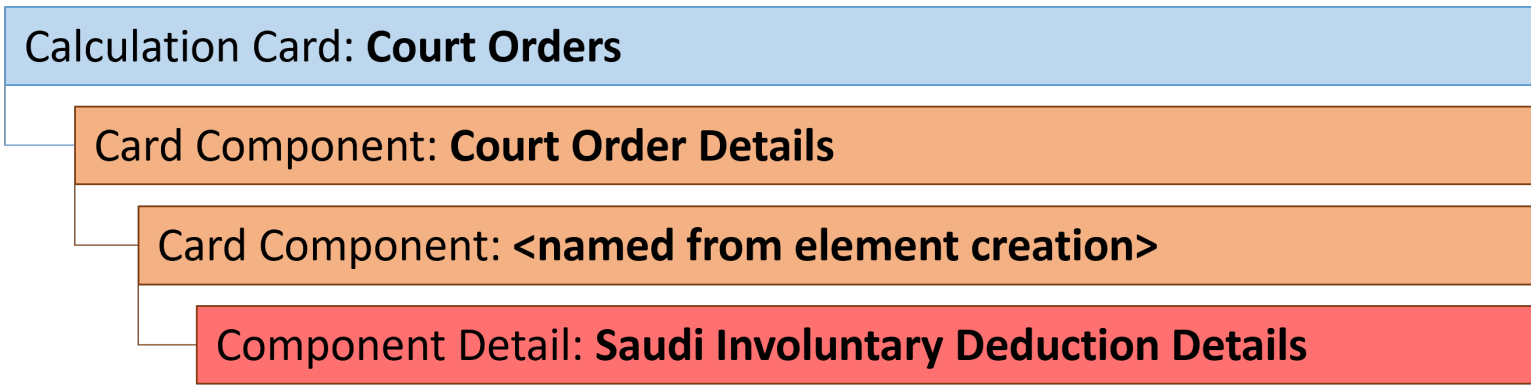
The Court Orders card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Court Orders card uses the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card comonoent.	ComponentDetail

Court Orders Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Court Orders is described in this diagram:



Related Topics

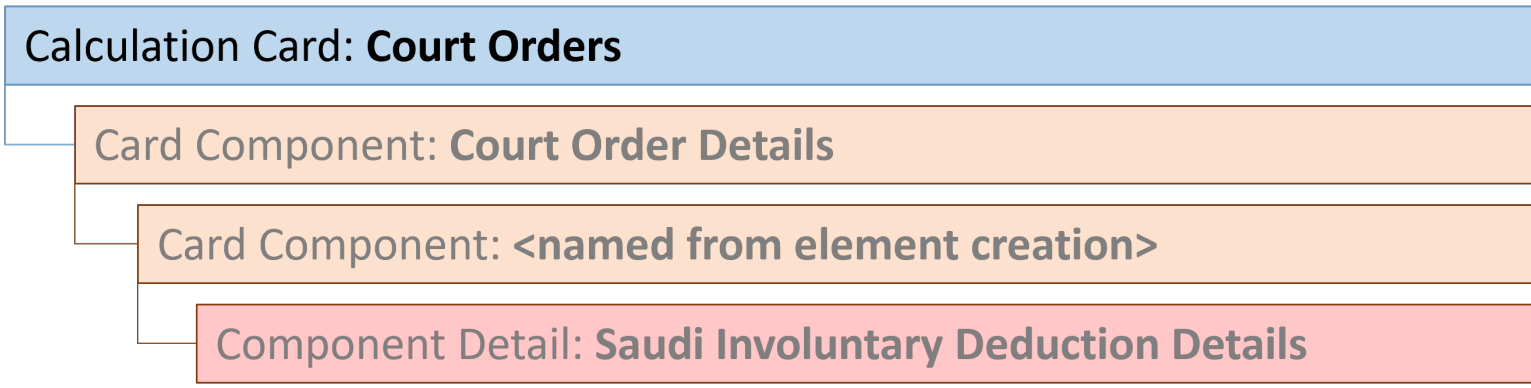
- Guidelines for Loading Calculation Cards
- Guidelines for Loading Court Orders for Saudi Arabia
- Guidelines for Loading Flexfield Data

Guidelines for Loading Court Orders for Saudi Arabia

Always include the Court Orders calculation card record, even if you're updating an existing Court Orders card.

Even if the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Court Orders Calculation Card Attributes



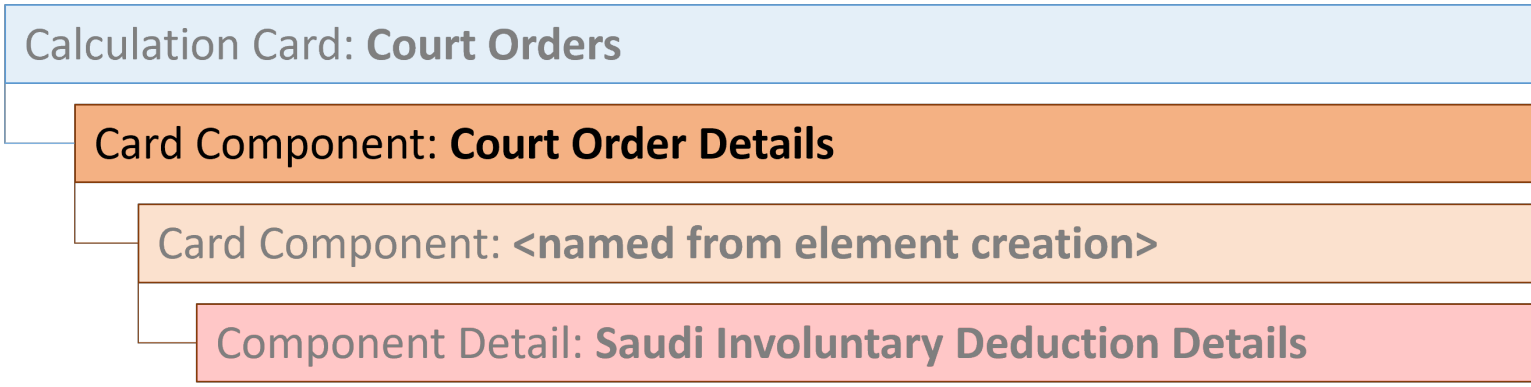
The Court Orders calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Court Orders calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Court Orders'.
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Court Order Details Card Component

The Court Order Details card component is a predefined card component, which acts as a parent to the element specific card component.



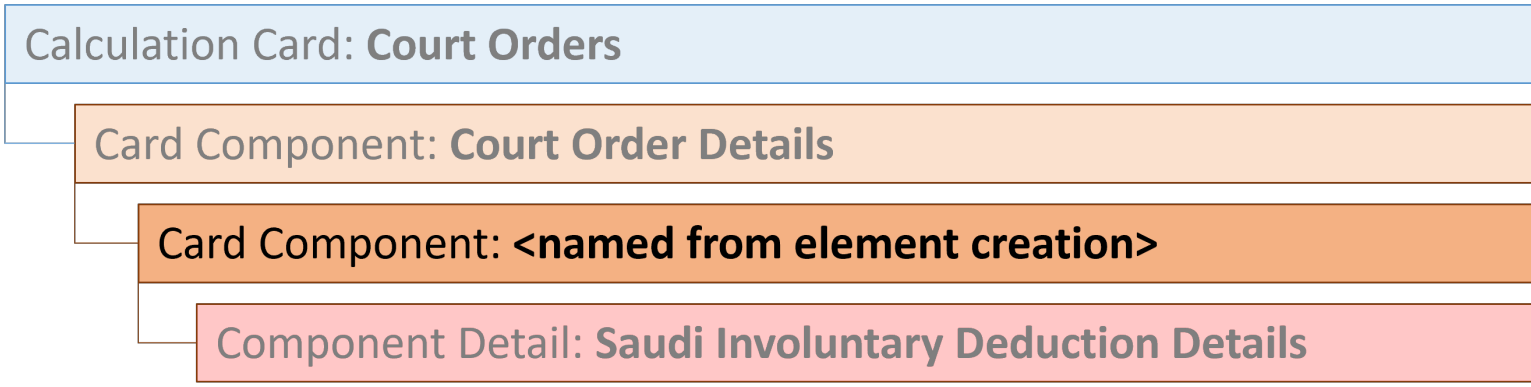
The Court Order Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Court Orders Details card component. For new card components, supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Court Order Details card component. This must be the same or after the EffectiveStartDate on the Court Orders calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Court Orders Details'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.

Card Component Named for the Court Order Element

This card component is named for the court orders element and is created as a child to the Court Order Details card component.



This card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the card component. For new card components,card supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders calculation should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
ParentDirCardCompId (SourceSystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	The parent Court Order Details card component. Use the same key type used to identify the Court Order Details card component.
EffectiveStartDate	N/A	The start date of the card component. This must be the same as the EffectiveStartDate on the Court Orders calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify the name defined when creating the court order element.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.
Context1	N/A	Reference code of the court order.

Component Detail Attributes for Court Orders

Calculation Card: **Court Orders**

Card Component: **Court Order Details**

Card Component: **<named from element creation>**

Component Detail: **Saudi Involuntary Deduction Details**

Court Orders uses a single flexfield context. Use the Component Detail record type to load the flexfield segment values.

Alongwith the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Saudi Involuntary Deduction Details (ORA_HRX_SA_INVOL_DED_DETAILS)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

To supply third-party details on the Saudi Involuntary Deduction Details flexfield context, refer to the Cloud Customer Connect topic *BI Publisher Report: Third Party Payees for Involuntary Deduction Cards* for a report that extracts the third-party names and party IDs for the country, legislation, and payee type specified.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent card component should be referenced using the same key type used to identify the parent record. This is the card component that is named after the Court Order element. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parentcard component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context. Supply ORA_HRX_SA_INVOL_DED_DETAILS. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Example of Creating Court Orders for Saudi Arabia
- Guidelines for Loading Flexfield Data
- Overview of Court Orders for Saudi Arabia

Example of Creating Court Orders for Saudi Arabia

This example creates a Court Orders card for employee assignment E191885.

The Saudi Involuntary Deduction Details flexfield context is populated with these values:

Flexfield Segment	Value
Amount	1000
Date of Issue	7 July 2020
Order Amount Payee Details	SA Court Order Payee

The CalculationCard.dat file is used to upload Court Orders calculation cards with HCM Data Loader:

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
AssignmentNumber|EffectiveStartDate
MERGE|CalculationCard|VISION|E191668_CO|SA LDG|Court Orders|E191668|2022/07/07

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
DirCardCompDefName|EffectiveStartDate|ParentDirCardCompId(SourceSystemId)|Context1
MERGE|CardComponent|VISION|E191668_COD|SA LDG|E191668_CO|Court Order Details|2022/07/07||
MERGE|CardComponent|VISION|E191668_SACO|SA LDG|E191668_CO|SA Court Order|2022/07/07|E191668_COD|REF_CO_3411

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|DirCardCompDefName|EffectiveStartDate|DirInformationCategory|FLEX:Deduction
 Developer DF|amount(Deduction Developer DF=ORA_HRX_SA_INVOL_DED_DETAILS)|dateOfIssue(Deduction
 Developer DF=ORA_HRX_SA_INVOL_DED_DETAILS)|orderAmountPayeeDetails_Display(Deduction Developer
 DF=ORA_HRX_SA_INVOL_DED_DETAILS)
MERGE|ComponentDetail|VISION|E191668_SACO_ID_DET|SA LDG|E191668_SACO|SA Court Order|2022/07/07|
ORA_HRX_SA_INVOL_DED_DETAILS|ORA_HRX_SA_INVOL_DED_DETAILS|1000.00|2022/07/07|SA Court Order Payee
```

32 Loading Payroll Localization Data for United Arab Emirates

Employee Social Insurance or Pension Fund Details

Overview of Employee Social Insurance Details for UAE

The Employee Social Insurance or Pension Fund Details card stores all information required to accurately compute social insurance contributions, such as employee details, annuity information and reference salary.

Considerations and Prerequisites

When the product license is set to Payroll or Payroll Interface, Employee Social Insurance or Pension Fund Details cards are automatically created when a new employee is entered using the New Hire task with a set of values specified at the person level, such as citizenship and social insurance information. The Social Insurance or Pension Fund Details card component is also auto-generated. You only populate the component details if you need to supply override values.

There may be cases where this information needs to be loaded in bulk:

During Data Migration:

Employee Social Insurance or Pension Fund Details information must be uploaded into Oracle Fusion Payroll, to ensure that contributions are calculated correctly. If HCM Data Loader is used to migrate employee records, a default Employee Social Insurance or Pension Fund Details card is automatically created. In most cases, the default won't reflect the employee's actual Social Insurance information, and therefore the card must be updated.

Ongoing Bulk Updates:

- When you bulk load New Hire information, a default Employee Social Insurance or Pension Fund Details card may be automatically generated (if the new hire records are created through HCM Data Loader or the interface with Taleo). In this case, you need to update the default card with the correct social insurance information.

It is recommended to have a good understanding of the Employee Social Insurance or Pension Fund Details card and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS for United Arab Emirates: Payroll Implementation and Functional Considerations (Document ID 1565449.1).

Employee Social Insurance or Pension Fund Details Calculation Card Record Types

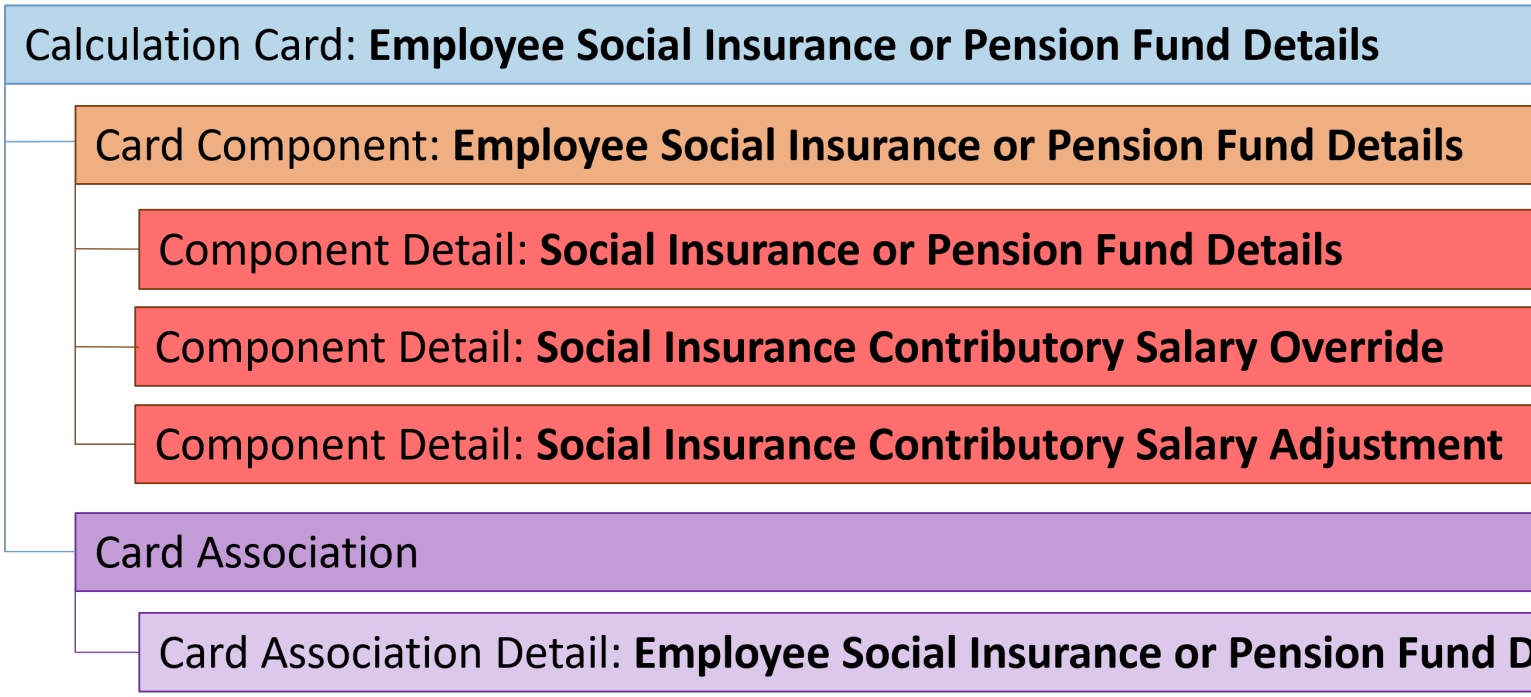
The Employee Social Insurance or Pension Fund Details card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Employee Social Insurance or Pension Fund Details card utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Employee Social Insurance or Pension Fund Details Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Employee Social Insurance or Pension Fund Details are described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Updating Employee Social Insurance Details for UAE
- Examples of Updating Employee Social Insurances Details for UAE
- Guidelines for Loading Flexfield Data

Guidelines for Updating Employee Social Insurance Details for UAE

You must have one Calculation Card record for every United Arab Emirates employee you are maintaining Social Insurance Detail for.

Even if you are updating an existing Employee Social Insurance Details card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Employee Social Insurance or Pension Fund Details Calculation Card Attributes

The Employee Social Insurance Details calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Employee Social Insurance or Pension Fund Details calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Employee Social Insurance or Pension Fund Details'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Employee Social Insurance or Pension Fund Details Card Component

The Employee Social Insurance or Pension Fund Details card component captures social insurance related information. It is created for you if the Employee Social Insurance or Pension Fund Details card is autogenerated.

If you need to update the defaulted social insurance information, provide an Employee Social Insurance or Pension Fund Details card component and related Component.

Card Component Attributes for Employee Social Insurance or Pension Fund Details Information

The Employee Social Insurance Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Employee Social Insurance or Pension Fund Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Social Insurance or Pension Fund Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Employee Social Insurance or Pension Fund Details card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Employee Social Insurance or Pension Fund Details calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
DirCardCompDefName	N/A	The component definition name. Specify 'Employee Social Insurance or Pension Fund Details'.
ComponentSequence	N/A	Specify '1'

Component Detail Attributes for Employee Social Insurance Details

Employee Social Insurance or Pension Fund Details card component uses three flexfield contexts. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- Social Insurance or Pension Fund Details (HRX_AE_SI_PREL)
- Social Insurance Contributory Salary Override (HRX_AE_SI_CONT_SAL_OVR_PREL)
- Social Insurance Contributory Salary Adjustment (HRX_AE_SI_CONT_SAL_ADJ_PREL)

Only the Social Insurance or Pension Fund Details context is required. The others are optional.

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Employee Social Insurance or Pension Fund Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Social Insurance or Pension Fund Details card component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_AE_SI_PREL'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Overview of Employee Social Insurance Details for UAE
- Examples of Updating Employee Social Insurances Details for UAE

Examples of Updating Employee Social Insurances Details for UAE

These examples update or maintain an auto-generated Employee Social Insurance or Pension Fund Details card.

User keys are used to identify the card and card component as the source keys may not be known for auto-generated cards.

Use the CalculationCard.dat file to update **Employee Social Insurance or Pension Fund Details** card component information with HCM Data Loader.

Example of Updating the Auto-Generated Employee Social Insurance or Pension Fund Details Card Component

This example updates the existing Employee Social Insurance or Pension Fund Details card component for an auto-generated Employee Social Insurance or Pension Fund Details card.

The Social Insurance or Pension Fund Details flexfield context is updated, setting these values

Flexfield Segment	Value
Citizenship	AE (Emirati)
Registered for Social Insurance or Pension	Y (Yes)
Date of Retirement	1 Sep 2025

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|AE LDG|Employee Social Insurance or Pension Fund Details|2019/01/01|E5252|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|AE LDG|Employee Social Insurance or Pension Fund Details|2019/01/01|E5252|1|Employee
 Social Insurance or Pension Fund Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer
 DF|_CITIZENSHIP(Deduction Developer DF=HRX_AE_SI_PREL)|_REGISTERED_FOR_SI(Deduction Developer
 DF=HRX_AE_SI_PREL)|_ORA_HRX_AE_RETIREMENT_DATE(Deduction Developer DF=HRX_AE_SI_PREL)
MERGE|ComponentDetail|AE LDG|Employee Social Insurance or Pension Fund Details|2019/01/01|E5252|1|Employee
 Social Insurance or Pension Fund Details|1|HRX_AE_SI_PREL|HRX_AE_SI_PREL|AE|Y|2025/09/01
```

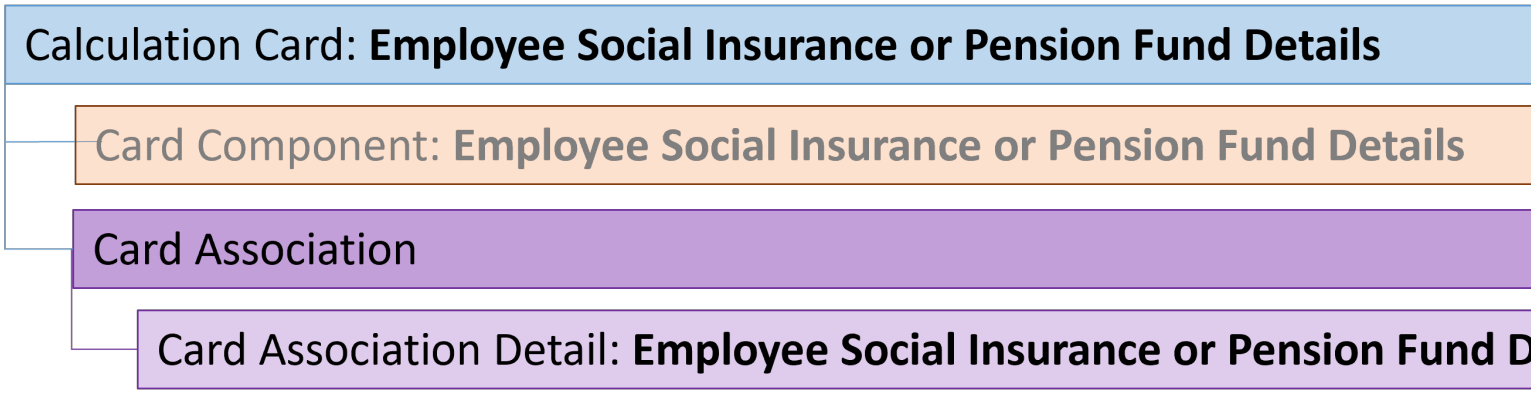
Example of Creating Social Insurance Contributory Salary Adjustment

This example populates the Social Insurance Contributory Salary Adjustment flexfield context for an auto-generated Employee Social Insurance or Pension Fund Details card. The Adjustment Amount is set to 5000.00.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
MERGE|CalculationCard|AE LDG|Employee Social Insurance or Pension Fund Details|2019/01/01|E5252|1

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|AE LDG|Employee Social Insurance or Pension Fund Details|2019/01/01|E5252|1|Employee
 Social Insurance or Pension Fund Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
_CONTRIBUTORY_SALARY_ADJUSTMENT(Deduction Developer DF=HRX_AE_SI_CONT_SAL_ADJ_PREL)
MERGE|ComponentDetail|AE LDG|Employee Social Insurance or Pension Fund Details|2019/01/01|E5252|1|Employee
 Social Insurance or Pension Fund Details|1|HRX_AE_SI_CONT_SAL_ADJ_PREL|HRX_AE_SI_CONT_SAL_ADJ_PREL|5000.00
```

Guidelines for Associating Employee Social Insurance Details Cards for UAE

The card association record associates the Employee Social Insurance or Pension Fund Details calculation card with the employee's tax reporting unit.



The associated tax reporting unit is defined in the card association. The card association detail allows the Employee Social Insurance or Pension Fund Details card component to be associated with the employee's assignments.

Card Association Attributes for Employee Social Insurance or Pension Fund Details

The Card Association record type associates the Employee Social Insurance or Pension Fund Details card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Employee Social Insurance or Pension Fund Details card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Employee Social Insurance or Pension Fund Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Social Insurance or Pension Fund Details calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Employee Social Insurance or Pension Fund Details calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Employee Social Insurance or Pension Fund Details

The card association detail associates the Employee Social Insurance or Pension Fund Details card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Thecard component this association is for. Supply either the source system ID value or user key attributes to identify the Employee Social Insurance or Pension Fund Details card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of Employee Social Insurance Details for UAE

Example of Loading Employee Social Insurance or Pension Fund Details Card Associations for UAE

This example creates the tax reporting unit association for the existing Employee Social Insurance or Pension Fund Details card belonging to employee assignment E2525.

User keys are provided to reference the existing Employee Social Insurance or Pension Fund Details card and Employee Social Insurance or Pension Fund Details card component on the assumption they were auto generated and therefore the source key values are unknown.

The CalculationCard.dat file is used to upload the Employee Social Insurance or Pension Fund Details card associations with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence
```

```
MERGE|CalculationCard|AE LDG|Employee Social Insurance or Pension Fund Details|2019/01/01|E2525|1

METADATA|CardAssociation|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|AssignmentNumber|
CardSequence|TaxReportingUnitName
MERGE|CardAssociation|AE LDG|Employee Social Insurance or Pension Fund Details|2019/01/01|E2525|1|AE Legal
 Entity

METADATA|CardAssociationDetail|LegislativeDataGroupName|DirCardDefinitionName|EffectiveStartDate|
AssignmentNumber|CardSequence|TaxReportingUnitName|DirCardCompDefName|ComponentSequence
MERGE|CardAssociationDetail|AE LDG|Employee Social Insurance or Pension Fund Details|2019/01/01|E2525|1|AE
 Legal Entity|Employee Social Insurance or Pension Fund Details|1
```

When associating an auto-generated card you don't need to include the CardComponent, unless also making changes to its values.

Example of Creating Employee Social Insurance or Pension Fund Details for UAE

This example creates an Employee Social Insurance or Pension Fund Details card with associations for employee assignment E5252.

If your Employee Social Insurance or Pension Fund Details cards are not auto-generated it is recommended that you use source keys to create the card.

The Social Insurance or Pension Fund Details flexfield context is populated with these values:

Flexfield Segment	Value
Citizenship	AE (Emirati)
Registered for Social Insurance or Pension	Y (Yes)
Date of Retirement	1 Sep 2025

The CalculationCard.dat file is used to upload Employee Social Insurance or Pension Fund Details calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|E5252_SI|AE LDG|Employee Social Insurance or Pension Fund Details|2019/01/01|
E5252

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|E5252_SI_DET|AE LDG|E5252_SI|2019/01/01|Employee Social Insurance or Pension Fund
 Details

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|FLEX:Deduction
 Developer DF|_CITIZENSHIP(Deduction Developer DF=HRX_AE_SI_PREL)|_REGISTERED_FOR_SI(Deduction Developer
 DF=HRX_AE_SI_PREL)|_ORA_HRX_AE_RETIREMENT_DATE(Deduction Developer DF=HRX_AE_SI_PREL)
MERGE|ComponentDetail|VISION|E5252_SI_PREL|AE LDG|E5252_SI_DET|2019/01/01|Employee Social Insurance or
 Pension Fund Details|HRX_AE_SI_PREL|HRX_AE_SI_PREL|AE|Y|2025/09/01

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|E5252_SI_TRU|AE LDG|E5252_SI|2019/01/01|AE Legal Entity
```

```
METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirRepCardId(SourceSystemId)|
DirCardCompId(SourceSystemId)|EffectiveStartDate|AssociationAssignmentNumber
MERGE|CardAssociationDetail|VISION|E5252_SI_TRU|E5252_SI_TRU|E5252_SI_DET|2019/01/01|E5252
```

End of Service Details

Overview of End of Service Details for UAE

The End of Service Details card stores all information required to accurately compute and process the gratuity payment, such as override amount and payment date.

The card is used to specify if the employee wants to claim the end of service remuneration or pension from the social insurance office.

Considerations and Prerequisites

When the product license is set to Payroll or Payroll Interface, End of Service Details cards are automatically created upon employee termination and the gratuity amount calculated. The Gratuity Details card component is also auto-generated along with the card association. You only populate the component details if you need to supply override values.

There may be cases where this information needs to be loaded in bulk:

During Data Migration:

End of Service Details information must exist in Oracle Fusion Payroll, to ensure that the gratuity is calculated correctly. If HCM Data Loader is used to migrate employee termination records, a default End of Service Details card is automatically created. For historical reasons you may wish to upload gratuity calculation cards to record the 'Latest Gratuity Payment Date', the last date a gratuity payment was made in a previous employment.

Ongoing Bulk Updates:

- If you perform mass upload of termination information, a default End of Service Details card may be generated automatically (if the termination records are created using HDL). In this case, you need to update the default card with the correct Gratuity information and create a component detail to capture the gratuity override amount, if required.

It is recommended to have a good understanding of the End of Service Details card and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS for United Arab Emirates: Payroll Implementation and Functional Considerations (Doc ID 1565449.1).

End of Service Details Calculation Card Record Types

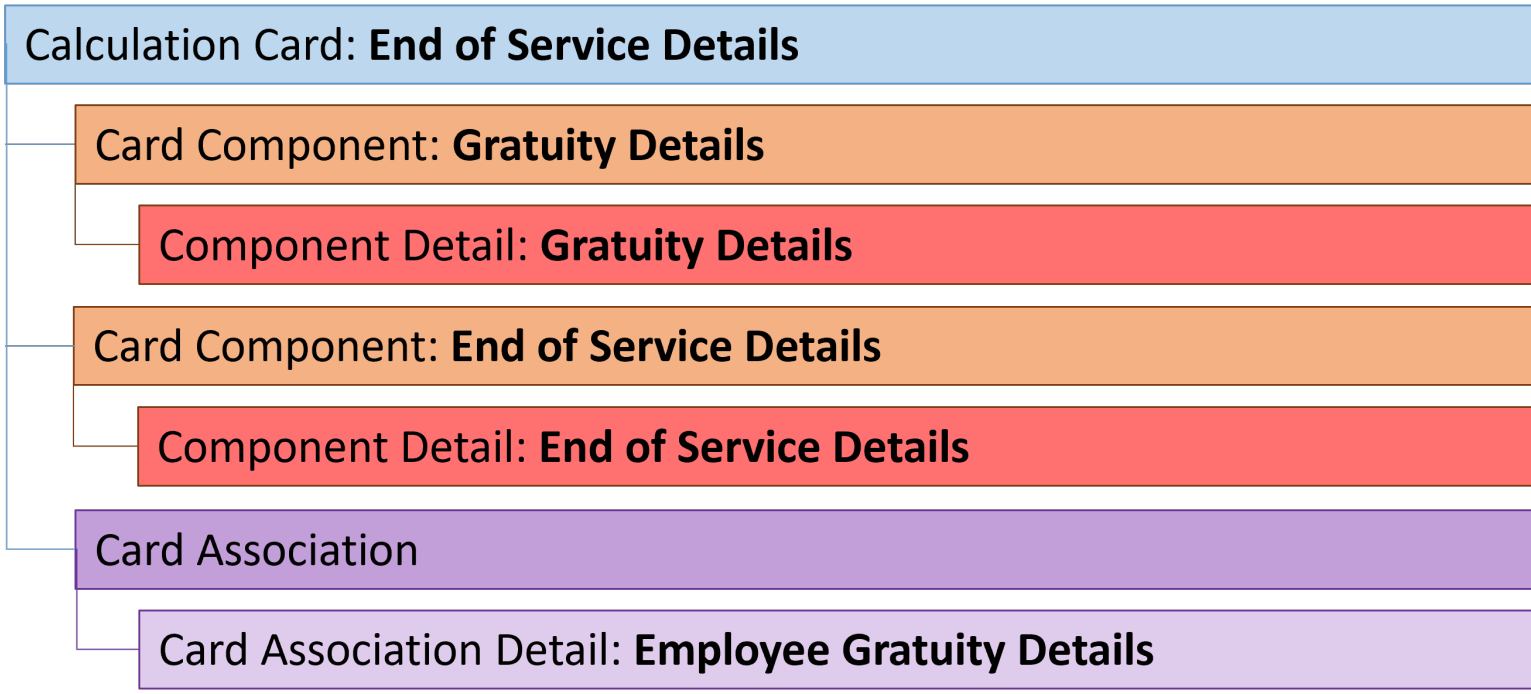
The End of Service Details card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The End of Service Details card utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

End of Service Details Calculation Card Hierarchy

The hierarchy of calculation card components applicable to End of Service Details are described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Updating End of Service Details for UAE
- Guidelines for Associating End of Service Details Cards for UAE
- Example of Creating End of Service Details for UAE
- Guidelines for Loading Flexfield Data

Guidelines for Updating End of Service Details for UAE

You must have one Calculation Card record for every terminated UAE employee you are maintaining End of Service details for.

Even if you are updating an existing End of Service Details card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

End of Service Details Calculation Card Attributes

The End of Service Details calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the End of Service Details calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'End of Service Details'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSequence	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Gratuity Details Card Component

The Gratuity Details card component captures the gratuity override amount and payment date information. It is created for you if the End of Service Details card is autogenerated. If you need to override the gratuity amount, provide a Gratuity Details card component and related Component Detail.

Calculation Card: **End of Service Details**

Card Component: **Gratuity Details**

Component Detail: **Gratuity Details**

Card Component Attributes for Gratuity Details

The Gratuity Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Gratuity Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent End of Service Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Gratuity Details card component, typically the employee's start date. This must be the same or later than the EffectiveStartDate on the End of Service Details calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Gratuity Details'.
ComponentSequence	N/A	Specify '1'

Component Detail Attributes for Gratuity Details

Gratuity Details card component uses one flexfield context. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

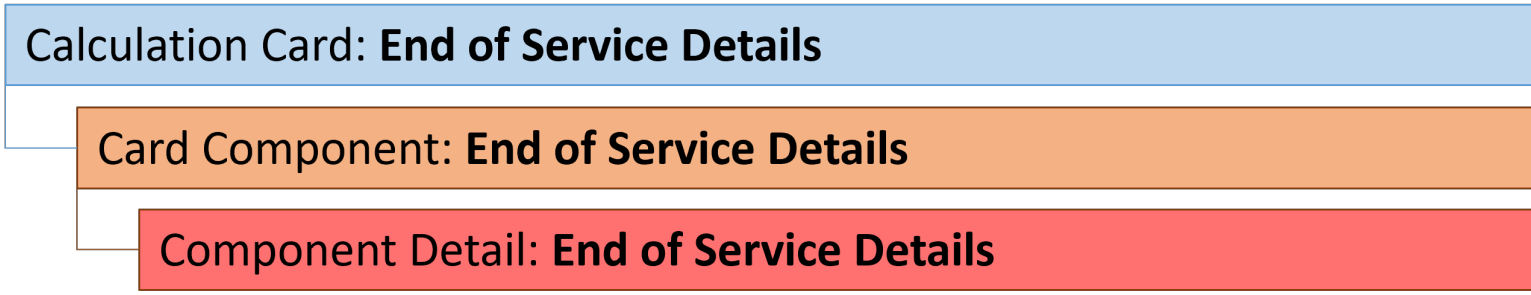
- Gratuity Details (HRX_AE_GRATUITY_PREL)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent Gratuity Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Gratuity Details card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_AE_GRATUITY_PREL'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

End of Service Details Card Component

The End of Service Details card component is mainly used for reporting Form 2 terminations and submitted to the social insurance office.



Card Component Attributes for End of Service Details

The End of Service Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the End of Service Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent End of Service Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the End of Service Details card component, typically the employee's start date. This must be the same or later than the EffectiveStartDate on the End of Service Details calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'End of Service Details'.
ComponentSequence	N/A	Specify '1'

Component Detail Attributes for End of Service Details

End of Service Details card component uses one flexfield context. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- End of Service Details (HRX_AE_EOS_PREL)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent End of Service Details card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the End of Service Details card component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_AE_EOS_PREL. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Overview of End of Service Details for UAE
- Examples of Updating End of Service Details for UAE

Examples of Updating End of Service Details for UAE

These examples update or maintain an auto-generated End of Service Details card.

User keys are used to identify the card and card component as the source keys may not be known for auto-generated cards.

Use the CalculationCard.dat file to update End of Service Details card component information with HCM Data Loader.

Example of Updating the Gratuity Details Card Component

This example updates the auto-generated Gratuity Details card component for the End of Service Details card.

The Gratuity Details flexfield context is updated, setting these values:

Flexfield Segment	Value
Override Amount	10050.00
Latest Gratuity Payment Date	31 Dec 2019

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber
MERGE|CalculationCard|AE LDG|End of Service Details|1|2019/09/01|E5252

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|AE LDG|End of Service Details|1|2019/12/15|E5252|Gratuity Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer DF|
_OVERRIDE_AMOUNT(Deduction Developer DF=HRX_AE_GRATUITY_PREL)|oraHrxAeLstGratPymtDate(Deduction Developer
 DF=HRX_AE_GRATUITY_PREL)
MERGE|ComponentDetail|AE LDG|End of Service Details|1|2019/12/15|E5252|Gratuity Details|1|
HRX_AE_GRATUITY_PREL|HRX_AE_GRATUITY_PREL|10050.00|2019/12/31
```

Example of Creating an End of Service Details Card Component

This example creates the End of Service Details card component and component details.

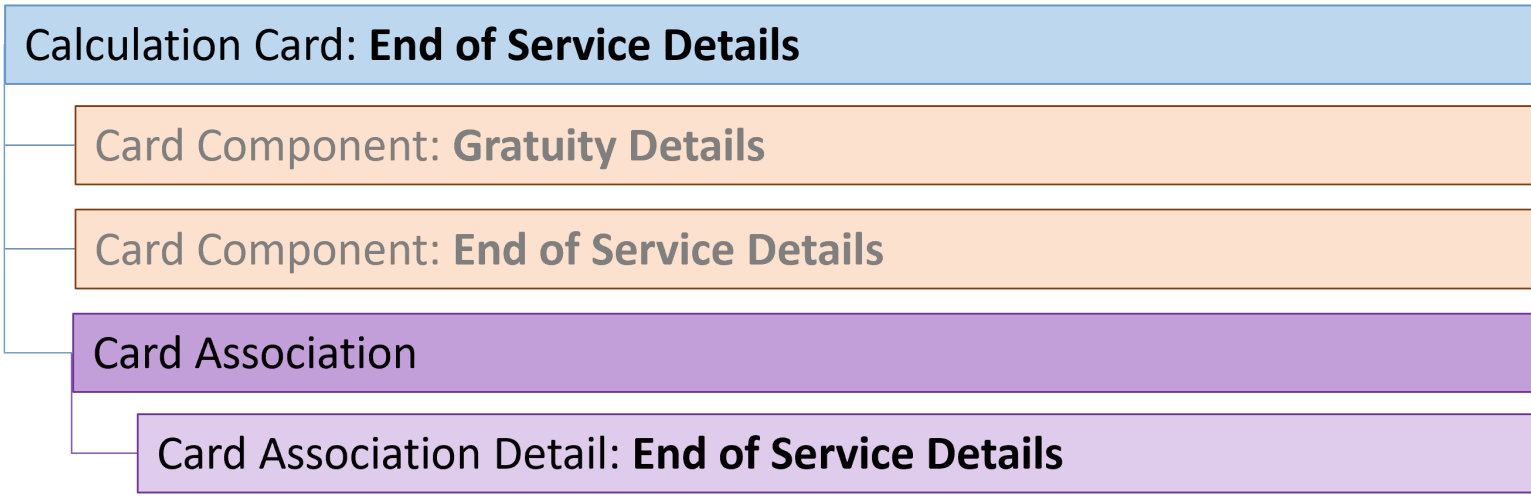
The End of Service Details flexfield context is updated, setting these values:

Flexfield Segment	Value
End of Service Remuneration or Pension?	Y (Yes)
Bank Name	United Arab Bank
Bank Branch	Dubai Almas Branch
Account Number	1234567890123456

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber
MERGE|CalculationCard|AE LDG|End of Service Details|1|2019/09/01|E5252

METADATA|CardComponent|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence
MERGE|CardComponent|AE LDG|End of Service Details|1|2019/12/15|E5252|End of Service Details|1

METADATA|ComponentDetail|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|DirCardCompDefName|ComponentSequence|DirInformationCategory|FLEX:Deduction Developer
 DF|_EOS_REMUNERATION_OR_PENSION(Deduction Developer DF=HRX_AE_EOS_PREL)|_BANK_NAME(Deduction Developer
 DF=HRX_AE_EOS_PREL)|_BANK_BRANCH(Deduction Developer DF=HRX_AE_EOS_PREL)|_ACCOUNT_NUMBER(Deduction
 Developer DF=HRX_AE_EOS_PREL)
MERGE|ComponentDetail|AE LDG|End of Service Details|1|2019/12/15|E5252|End of Service Details|1|
HRX_AE_EOS_PREL|HRX_AE_EOS_PREL|Y|United Arab Bank|Dubai Almas Branch|1234567890123456
```

Guidelines for Associating End of Service Details Cards for UAE

The card association record associates the End of Service Details calculation card with the employee's tax reporting unit.



The associated tax reporting unit is defined in the card association. The card association detail allows the Gratuity Details card component to be associated with the employee's assignments.

Card Association Attributes for End of Service Details

The Card Association record type associates the End of Service Details card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the End of Service Details card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent End of Service Details calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the End of Service Details calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the End of Service Details calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for End of Service Details

The card association detail associates the Gratuity Details card component with the payroll assignments for the employee. If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	Thecard component this association is for. Supply either the source system ID value or user key attributes to identify the Gratuity Details card component.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Related Topics

- Overview of End of Service Details for UAE

Example of Loading End of Service Details Card Associations for UAE

This example creates the tax reporting unit association for the existing End of Service Details card belonging to employee assignment E5252.

User keys are provided to reference the existing End of Service Details card and Gratuity Details card component on the assumption they were auto generated and therefore the source key values are unknown.

The CalculationCard.dat file is used to upload the End of Service Details card associations with HCM Data Loader.

```
METADATA|CalculationCard|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber
MERGE|CalculationCard|AE LDG|End of Service Details|1|2019/09/01|E5252

METADATA|CardAssociation|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|EffectiveStartDate|
AssignmentNumber|TaxReportingUnitName
MERGE|CardAssociation|AE LDG|End of Service Details|1|2019/12/15|E2525|AE Legal Entity

METADATA|CardAssociationDetail|LegislativeDataGroupName|DirCardDefinitionName|CardSequence|
EffectiveStartDate|AssignmentNumber|TaxReportingUnitName|DirCardCompDefName|ComponentSequence|
AssociationAssignmentNumber
MERGE|CardAssociationDetail|AE LDG|End of Service Details|1|2019/12/15|E2525|AE Legal Entity|Gratuity
 Details|1|E5252
```

When associating an auto-generated card you don't need to include the CardComponent unless also making changes to its values.

Example of Creating End of Service Details for UAE

This example creates an End of Service Details card with associations for employee assignment E5252.

If your End of Service Details cards are not auto-generated it is recommended that you use source keys to create the card.

The Gratuity Details flexfield context is populated with these values:

Flexfield Segment	Value
Override Amount	10050.00
Latest Gratuity Payment Date	31 Dec 2019

The End of Service Details context is populated with these values:

Flexfield Segment	Value
End of Service Remuneration or Pension?	Y (Yes)
Bank Name	United Arab Bank
Bank Branch	Dubai Almas Branch
Account Number	1234567890123456

The CalculationCard.dat file is used to upload End of Service Details calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
EffectiveStartDate|AssignmentNumber
MERGE|CalculationCard|VISION|E5252_EOS|AE LDG|End of Service Details|2022/09/01|E5252

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
EffectiveStartDate|DirCardCompDefName
MERGE|CardComponent|VISION|E5252_EOS_GD|AE LDG|E5252_EOS|2022/09/01|Gratuity Details
MERGE|CardComponent|VISION|E5252_EOS_EOS|AE LDG|E5252_EOS|2022/09/01|End of Service Details

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|EffectiveStartDate|DirCardCompDefName|DirInformationCategory|
FLEX:Deduction Developer DF|_OVERRIDE_AMOUNT(Deduction Developer DF=HRX_AE_GRATUITY_PREL)|
oraHrxAeLstGratPymtDate(Deduction Developer DF=HRX_AE_GRATUITY_PREL)|_EOS_REMUNERATION_OR_PENSION(Deduction
 Developer DF=HRX_AE_EOS_PREL)|_BANK_NAME(Deduction Developer DF=HRX_AE_EOS_PREL)|_BANK_BRANCH(Deduction
 Developer DF=HRX_AE_EOS_PREL)|_ACCOUNT_NUMBER(Deduction Developer DF=HRX_AE_EOS_PREL)
MERGE|ComponentDetail|VISION|E5252_EOS_GD_CD|AE LDG|E5252_EOS_GD|2022/09/01|Gratuity Details|
HRX_AE_GRATUITY_PREL|HRX_AE_GRATUITY_PREL|20500.00|2019/09/30||||
MERGE|ComponentDetail|VISION|E5252_EOS_EOS_CD|AE LDG|E5252_EOS_EOS|2022/09/01|End of Service Details|
HRX_AE_EOS_PREL|HRX_AE_EOS_PREL|||Y|United Arab Bank|Dubai Almas Branch|1234567890123456

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardId(SourceSystemId)|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|E5252_EOS_AE|AE LDG|E5252_EOS|2022/09/01|AE Legal Entity

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirRepCardId(SourceSystemId)|EffectiveStartDate|DirCardCompId(SourceSystemId)|AssociationAssignmentNumber
```

```
MERGE|CardAssociationDetail|VISION|E5252_EOS_AED|AE LDG|E5252_EOS_AE|2022/09/01|E5252_EOS_GD|E5252
```

Court Orders

Overview of Court Orders for UAE

The Court Orders card stores information for court order deductions, such as amount and payee.

Considerations and Prerequisites

Use the Elements task to create Court Order elements. When you create court order elements, a card component definition is generated with the same name as the element. The generated card component definition is then available to use on the predefined Court Orders calculation card.

Unlike Employee Social Insurance or Pension Fund Details or End of Service Details cards, the Court Order card is not generated automatically. You only need to create this card for those employees who are subject to court order deductions.

There may be cases when there is a need to bulk load Court Orders cards:

During Data Migration:

Employee's Court Orders must bemigrated to the Oracle Payroll Cloud to ensure that appropriate deductions are consider. Using HCM Data Loader you must create the card and all necessary card components and component details.

Ongoing Bulk Updates:

- New hires may have Court Orders that need to be uploaded in bulk.

It is recommended to have a good understanding of the Court Orders card and the information it contains prior to attempting mass upload. For further information, see **Oracle Fusion HRMS for United Arab Emirates: Payroll Implementation and Functional Considerations (Doc ID 15654491)**.

Before Bulk Loading Court Order Information:

- Create eligibility for the Court Order element.
- Create third-party payee and third-party payment methods for the people and organizations the payment of the court order is made to.
- Update the PAY_INVLN_DEDN_ORDER_AMOUNT_PAYEE lookup type with the court order issuing authorities.

Court Order Calculation Card Record Types

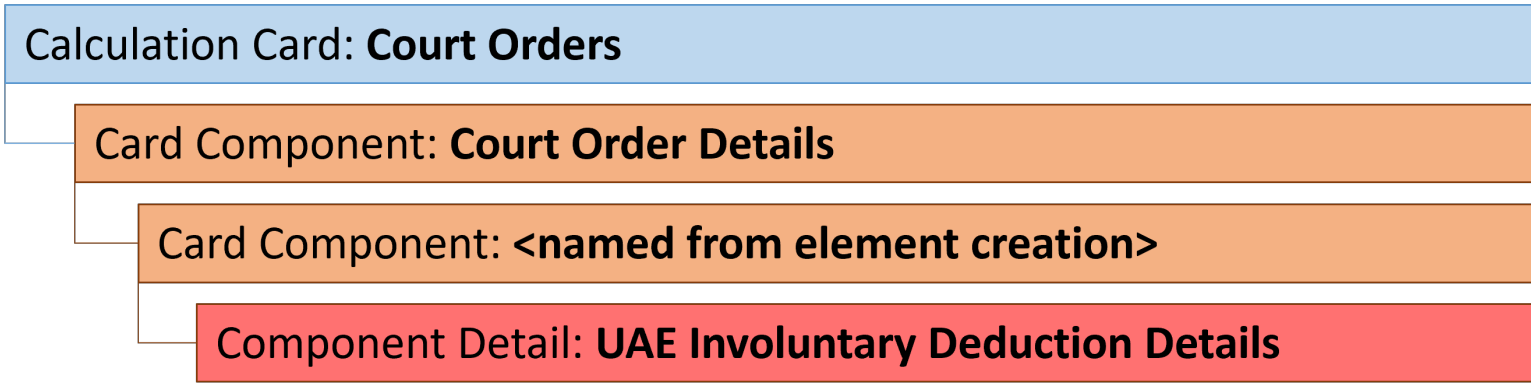
The Court Orders card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

The Court Orders card utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card component.	ComponentDetail

Court Orders Calculation Card Hierarchy

The hierarchy of calculation card components applicable to Court Orders are described in this diagram:



Related Topics

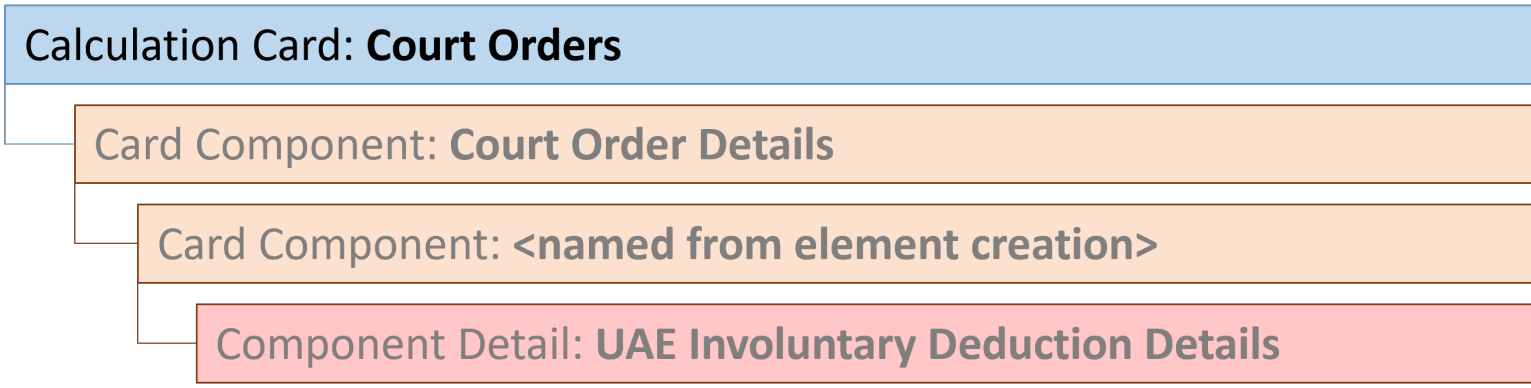
- Guidelines for Loading Calculation Cards
- Guidelines for Loading Court Orders for UAE
- Guidelines for Loading Flexfield Data

Guidelines for Loading Court Orders for UAE

Always include the Court Orders calculation card record, even if you're updating an existing Court Orders card.

Even if the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Court Orders Calculation Card Attributes



The Court Orders calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Court Orders calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Court Orders'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. **Note:** Not required when source keys are used.

Court Order Details Card Component

The Court Order Details card component is a predefined card component which acts as a parent to the element specific card component.

Calculation Card: **Court Orders**

Card Component: **Court Order Details**

Card Component: **<named from element creation>**

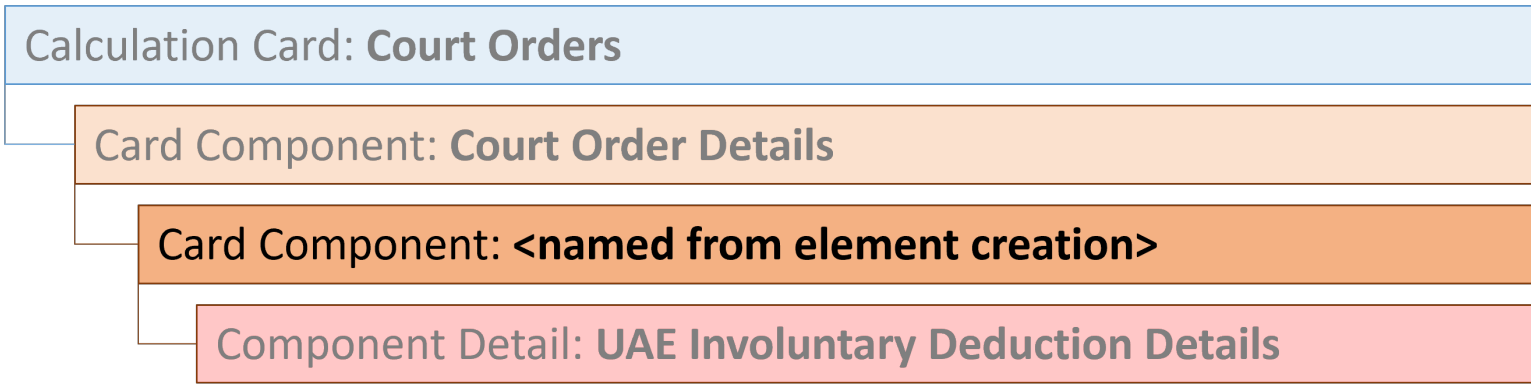
Component Detail: **UAE Involuntary Deduction Details**

The Court Order Details card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the Court Orders Details card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the Court Order Details card component. This must be the same or after the EffectiveStartDate on the Court Orders calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify 'Court Orders Details'.
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. **Note:** Not required when source keys are used.

Card Component Named for the Court Order Element

This card component is a named for the court orders element and is created as a child to the Court Order Details card component.

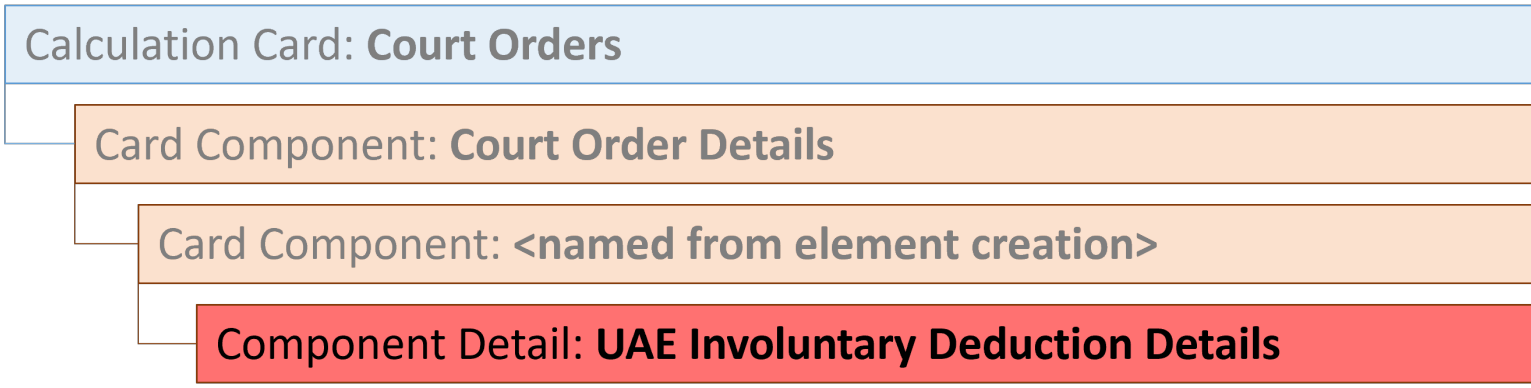


This card component uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Court Orders calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
ParentDirCardCompId (SourceSystemId)	CardSequence, ParentComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, ParentDirCardCompDefName	The parent Court Order Details card component. Use the same key type used to identify the Court Order Details card component.
EffectiveStartDate	N/A	The start date of the card component. This must be the same as the EffectiveStartDate on the Court Orders calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify the name defined when creating the court order element.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
ComponentSequence	N/A	A number to uniquely identify this card component when multiple card components for the same card component definition exist on the same card. Not required when source keys are used.
Context1	N/A	Reference code of the court order.

Component Detail Attributes for Court Orders



Court Orders use a single flexfield context. Use the Component Detail record type to load the flexfield segment values.

In addition to the attributes defined here, include the necessary flexfield segment attribute values for these flexfield contexts:

- UAE Involuntary Deduction Details (ORA_HRX_AE_INVOL_DED_DETAILS)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

To supply third-party details on the UAE Involuntary Deduction Details flexfield context, refer to the Cloud Customer Connect topic *BI Publisher Report: Third Party Payees for Involuntary Deduction Cards* for a report that extracts the third-party names and party IDs for the country, legislation and payee type specified.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName,	The parent card component should be referenced using the same key type used to

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
	LegislativeDataGroupName, DirCardCompDefName	identify the parent record. This is the card component that is named after the Court Order element. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the parentcard component.
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context. Supply ORA_HRX_SA_INVOL_DED_DETAILS. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Court Orders for UAE
- Example of Creating Court Orders for UAE
- Guidelines for Loading Flexfield Data

Example of Creating Court Orders for UAE

This example creates a Court Orders card for employee assignment E5252.

The UAE Involuntary Deduction Details flexfield context is populated with these values

Flexfield Segment	Value
Amount	1000
Date of Issue	7 July 2020

Flexfield Segment	Value
Order Amount Payee Details	AE Court Order Payee

The CalculationCard.dat file is used to upload Court Orders calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardDefinitionName|
AssignmentNumber|EffectiveStartDate
MERGE|CalculationCard|VISION|E5252_CO|AE LDG|Court Orders|E5252|2019/07/07

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|DirCardId(SourceSystemId)|
DirCardCompDefName|EffectiveStartDate|Context1|ParentDirCardCompId(SourceSystemId)
MERGE|CardComponent|VISION|E5252_COD|AE Regression_LDG|E5252_CO|Court Order Details|2019/07/07||
MERGE|CardComponent|VISION|E5252_AECO|AE LDG|E5252_CO|PM AE Court Order|2019/07/07|7777|E5252_COD

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|LegislativeDataGroupName|
DirCardCompId(SourceSystemId)|DirCardCompDefName|EffectiveStartDate|DirInformationCategory|FLEX:Deduction
 Developer DF|amount(Deduction Developer DF=ORA_HRX_AE_INVOL_DED_DETAILS)|dateOfIssue(Deduction
 Developer DF=ORA_HRX_AE_INVOL_DED_DETAILS)|orderAmountPayeeDetails_Display(Deduction Developer
 DF=ORA_HRX_AE_INVOL_DED_DETAILS)
MERGE|ComponentDetail|VISION|E5252_AECO_ID_DET|AE Regression_LDG|E5252_AECO|PM AE Court Order|2019/07/07|
ORA_HRX_AE_INVOL_DED_DETAILS|ORA_HRX_AE_INVOL_DED_DETAILS|1000.00|2020/07/07|AE Court Order Payee
```

33 Loading Payroll Localization Data for Australia

Overview of Statutory Deductions for Australia

Statutory Deduction cards store all the information required to accurately compute tax, superannuation and payroll tax.

One card must be created for each employee and for the TRU that the employee reports to.

Considerations and Prerequisites

If the product license is set to Payroll or Payroll Interface, employee Statutory Deduction cards are automatically created when a new employee is entered using the New Hire task with a set of default values specified at the Payroll Statutory Unit (PSU) or the Tax Reporting Unit (TRU) level.

However, there may be cases where this information must be loaded in bulk:

During data migration:

Employee PAYG and superannuation information must be uploaded into Oracle Fusion Payroll, to ensure that contributions are calculated correctly. If HCM Data Loader is used to migrate employee records, a default statutory deduction card is automatically created. The default may not reflect the employee's actual tax and superannuation information, and therefore the card must be updated.

Ongoing bulk updates:

- Bulk loading of new hire data: If you have to do a mass upload of New Hire information, a default Statutory Deductions card may be automatically generated (if the new hire records are created through HCM Data Loader or the interface with Taleo). In this case, you need to update the default card with the correct information.

You must create element eligibility for element Statutory Deductions. This needs to be done once for each LDG that is used for entering person calculation cards.

It is recommended to have a good understanding of the Statutory Deductions card and the information it contains prior to attempting mass upload as it has a direct impact on statutory deductions and reporting. For further information, see Oracle Fusion HRMS (Australia): HR Implementation and Functional Considerations Technical Brief (Doc ID 1628365.1).

Statutory Deduction Card Record Types

The Statutory Deductions card is uploaded with HCM Data Loader using the Global Payroll Calculation Card business object. This generic object hierarchy provides record types to support the various country-specific requirements.

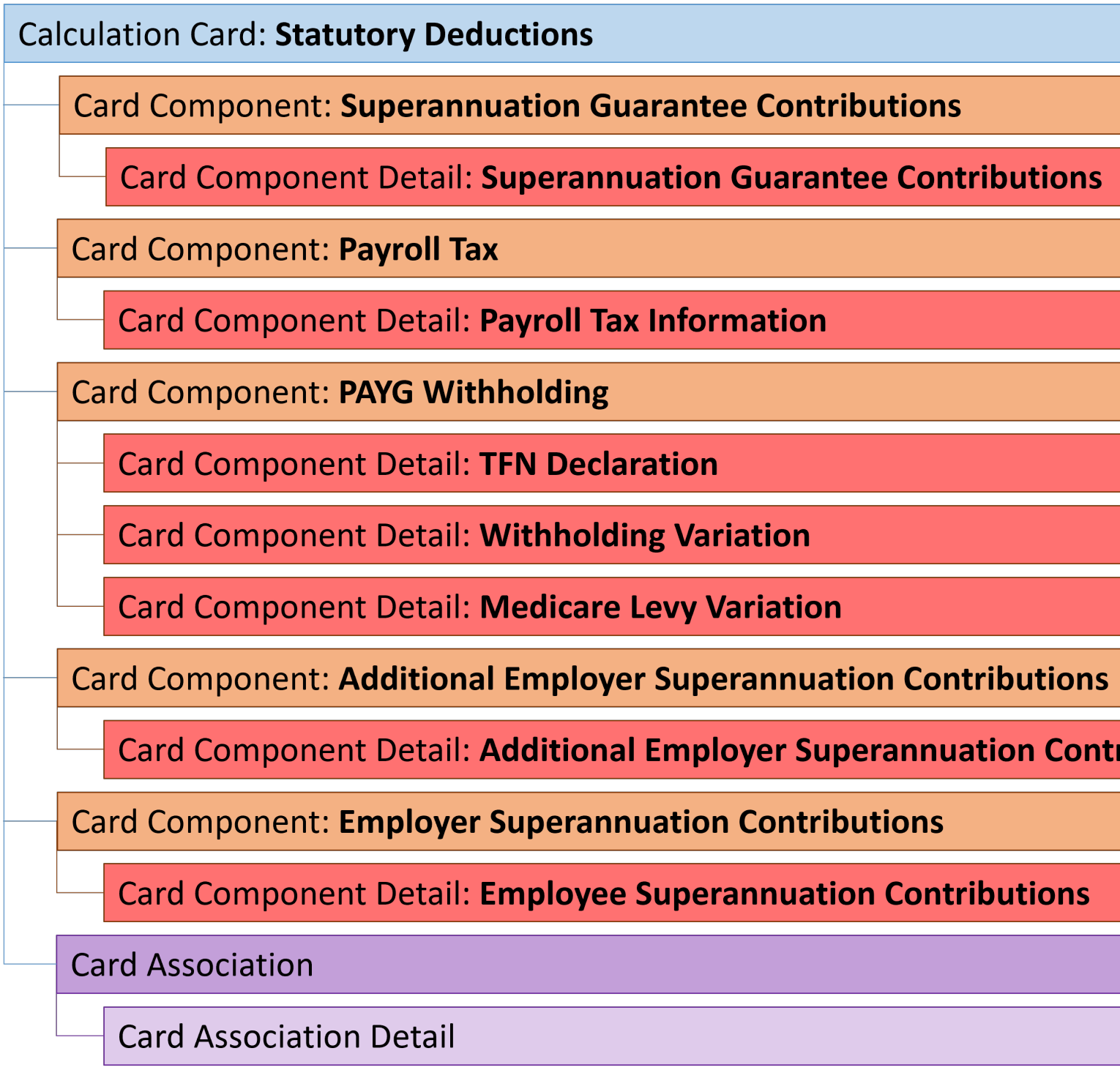
The French Statutory Deductions utilizes the following record types:

Component	Functional Description	File Discriminator
Calculation Card	Defines the calculation card type and the employee assignment that it captures information for.	CalculationCard

Component	Functional Description	File Discriminator
Card Component	Used to group and segregate data required by the calculation card. The following sections describe the card components applicable to this calculation card and the child records that are required for each card component.	CardComponent
Component Detail	Provide a component detail record for each flexfield context required by each card comonoent.	ComponentDetail
Card Association	Associates the calculation card with the Tax Reporting Unit the employee reports to.	CardAssociation
Card Association Detail	Associates card components with the employee's assignments.	CardAssociationDetails

Statutory Deduction Calculation Card Hierarchy

The hierarchy of calculation card components applicable Statutory Deductions are described in this diagram:



Related Topics

- Guidelines for Loading Calculation Cards
- Guidelines for Loading Statutory Deductions for Australia

Guidelines for Loading Statutory Deductions for Australia

You must have one Calculation Card record for every employee you are maintaining statutory deduction data for.

Even if you are updating an existing Statutory Deduction card and the calculation card itself isn't being updated, you must still include the calculation card record to group other related data supplied in the file.

Statutory Deductions Calculation Card Attributes

The Statutory Deductions calculation card uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	A unique identifier for the Statutory Deductions calculation card. For new calculation card supply the source key attributes. You can also identify calculation cards with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the calculation card.
DirCardDefinitionName	N/A	The name of the card definition. Specify 'Statutory Deductions'
AssignmentId(SourceSystemId)	AssignmentNumber	Either supply the source system ID, or assignment number that identifies the employee assignment this calculation card is for.
EffectiveStartDate	N/A	The start date of the calculation card, typically the employee's start date.
CardSeqeunce	N/A	A number to uniquely identify this card when multiple calculation cards for the same card definition exist for the employee. Not required when source keys are used.

Statutory Deduction Card Components

The Statutory Deductions card supports the following card components:

- Superannuation Guarantee Contributions
- Payroll Tax
- PAYG Withholding
- Additional Employer Superannuation Contributions

- Employer Superannuation Contributions

The attributes to supply are similar for each card component, the only difference being the value to supply to the DirCardCompDefName attribute which accepts the card component name.

Card Component Attributes

The card component record uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	A unique identifier for the card component. For new card components supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card.
EffectiveStartDate	N/A	The start date of the card component, typically the employee's start date. This must be the same as the EffectiveStartDate on the Statutory Deductions calculation card.
EffectiveEndDate	N/A	The end date is optional for the card component
DirCardCompDefName	N/A	The component definition name. Specify one of: • Superannuation Guarantee Contributions • Payroll Tax • PAYG Withholding • Additional Employer Superannuation Contributions • Employer Superannuation Contributions
ComponentSequence	N/A	Specify '1'.

Card Component Detail Attributes

In addition to the attributes defined here, include the necessary flexfield segment attribute values for the flexfield contexts supported by each card component:

Card Component: Superannuation Guarantee Contributions

- Superannuation Guarantee Contributions(HRX_AU_SG_CONTRIBUTIONS)

Card Component: Payroll Tax

- Payroll Tax Information (HRX_AU_PAYROLL_TAX)

Card Component: PAYG Withholding

- TFN Declaration (HRX_AU_TFN_DECLARATION)
- Withholding Variation (HRX_AU_PAYGW_VARIATION)
- Medicare Levy Variation (HRX_AU_MEDICARE_LEVY_VARIATION)

Card Component: Additional Employer Superannuation Contributions

- Additional Employer Superannuation Contributions (HRX_AU_ADDL_EMPLOYER_SUPER_CONTRIBUTIONS)

Card Component: Employer Superannuation Contributions

- Employer Superannuation Contributions (HRX_AU_PERSONAL_SUPER_CONTRIBUTIONS)

Note: You can find the flexfield segment attribute names for this flexfield context using the View Business Objects task.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, ComponentSequence, DirInformationCategory, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName, DirCardCompDefName	A unique identifier for the component detail. For new component detail records supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card component definition.
DirCardCompId(SourceSystemId)	CardSequence, ComponentSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName, DirCardCompDefName	The parent card component should be referenced using the same key type used to identify the parent record. When using source keys, supply this attribute with the value supplied for the card component's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card component.
EffectiveStartDate	N/A	The start date of the component detail, or update to the component detail if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Employee Information card component.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
EffectiveEndDate	N/A	The optional end date of the component detail, or if you are providing date-effective history the last day of the date-effective changes.
DirCardCompDefName	N/A	The name of the card component this detail is for. This is used to identify the flexfield context and should be supplied even when a source key is used to identify the parent card component.
DirInformationCategory	N/A	The code for the flexfield context, such as 'HRX_AU_SG_CONTRIBUTIONS'. Flexfield context codes are visible on the Flexfield Attributes tab of the View Business Objects task.
FLEX:Deduction DeveloperDF	N/A	Supply the same value as for the DirInformationCategory attribute.

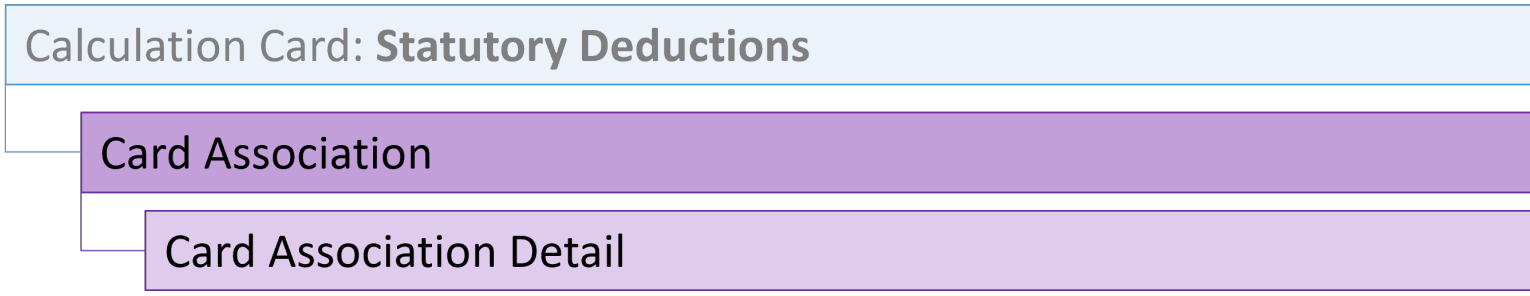
Related Topics

- Guidelines for Loading Calculation Cards
- Overview of Statutory Deductions for Australia
- Guidelines for Loading Flexfield Data

Guidelines for Loading Statutory Deductions Card Associations for Australia

The card association record associates the Statutory Deductions calculation card with a tax reporting unit.

If creating a Statutory Deductions card from scratch you can provide the association details with the Statutory Deductions card.



The associated tax reporting unit is defined in the card association. The card association details records allow the Employee Information card component to be associated with the employee assignments.

Card Association Attributes for Statutory Deductions

The Card Association record type associates a Statutory Deductions card with the employee's tax reporting unit.

The Card Association record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the Statutory Deductions card association. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirCardId(SourceSystemId)	CardSequence, AssignmentNumber, DirCardDefinitionName, LegislativeDataGroupName	The parent Statutory Deductions calculation card should be identified by using the same key type used to identify the calculation card. When using source keys, supply this attribute with the value supplied for the calculation card's SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent calculation card
EffectiveStartDate	N/A	The start date of the card association or the update to the card association if you are providing date-effective history. This must be after or equal to the EffectiveStartDate provided for the Tax Withholding calculation card.
TaxReportingUnitName	N/A	The name of the tax reporting unit to associate with the Tax Withholding calculation card. If source keys are used to identify the card association, you must still specify the LegislativeDataGroupName attribute to identify the tax reporting unit.

Card Association Detail Attributes for Statutory Deductions

The card association details record associates the Statutory Deductions card with the payroll assignments for the employee.

If the employee has more than one payroll assignment provide a card association detail record for each payroll assignment.

The Card Association Detail record type uses these attributes:

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
SourceSystemId	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	A unique identifier for the card association detail record. For new card associations supply the source key attributes. You can also identify card components with the user key attributes.
SourceSystemOwner	N/A	The name of the source system owner used to generate the source system ID.

HCM Data Loader Attribute	Alternative User Key Attributes	Functional Description
LegislativeDataGroupName	N/A	The name of the legislative data group for the card definition.
DirRepCardId(SourceSystemId)	CardSequence, TaxReportingUnitName, LegislativeDataGroupName, AssignmentNumber, DirCardDefinitionName	The parent card association record should be identified by using the same key type supplied to against the card association. When using source keys, supply this attribute with the value supplied for the card's association SourceSystemId attribute. Otherwise, supply the user key attributes with the same values as the parent card association.
EffectiveStartDate	N/A	The effective start date of the card association.
EffectiveEndDate	N/A	The effective end date of the card association.
AssociationAssignmentId(SourceSystemId)	AssociationAssignmentNumber	Identify the employee's assignment to associate with card component.

Example of Creating a New Statutory Deductions Card for Australia

This example creates a complete Statutory Deductions card with associations for assignment E17062.

It creates these components, populating the following fields:

Superannuation Guarantee Contributions

Flexfield Context: Superannuation Guarantee Contributions (HRX_AU_SG_CONTRIBUTIONS)

Attribute	Value
Super Fund Id	AU-Super Fund
Date Valid Choice Accepted	17-Jun-2020
Date Valid Choice Commenced	17-Jun-2020
Member Number	22JUN202301
Override Maximum Contribution Base	Y
Override Minimum Hours	Y
Override Minimum Age	Y
Override Monthly Minimum Earnings	Y
SGC Under 18 Hours	7

Employee Superannuation Contributions

Flexfield Context: Employee Superannuation Contributions (HRX_AU_PERSONAL_SUPER_CONTRIBUTIONS)

Attribute	Value
Super Fund Id	AU-Super Fund
Member Number	123456
Pretax Deduction	Y

Additional Employer Superannuation Contributions

Flexfield Context: Additional Employer Superannuation Contributions (HRX_AU_ADDL_EMPLOYER_SUPER_CONTRIBUTIONS)

Attribute	Value
Super Fund Id	AU-Super Fund
Member Number	231231001
Reportable Employer Super Contribution	Y
Contribution Type	Voluntary Amount (ORA_VOLUNTARY)

PAYG Withholding

Flexfield Context: TFN Declaration (HRX_AU_TFN_DECLARATION)

Attribute	Value
Australian Resident	N
Declaration Date	17-Jun-2020
STSL	N
Basis of Payment	Full Time (FULL)
Senior Australian	Single (SINGLE)
Submitted to ATO	17-Jun-2020
Tax Free Threshold	N
Tax Offset Amount	50
Tax File Number	110987566
WHM Country Code	India
Working Holiday Maker	Y

Flexfield Context: Withholding Variation (HRX_AU_PAYGW_VARIATION)

Attribute	Value
Include Additional Payments	N
Variation Type	Increase (INCREASE)

Flexfield Context: Medicare Levy Variation (HRX_AU_MEDICARE_LEVY_VARIATION)

Attribute	Value
Exemption Type	Full (FULL)
Income Less Than Threshold	N
Dependent Children	2
Spouse	N
Medicare Levy Surcharge	1.25% (0.0125)

Payroll Tax

Flexfield Context : Payroll Tax Information (HRX_AU_PAYROLL_TAX)

Attribute	Value
Exclude from Payroll Tax	N
State	VIC

The card is associated with the AU TRU tax reporting unit.

The CalculationCard.dat file is used to upload Statutory Deductions calculation cards with HCM Data Loader.

```
METADATA|CalculationCard|SourceSystemOwner|SourceSystemId|DirCardDefinitionName|LegislativeDataGroupName|
EffectiveStartDate|AssignmentNumber|LegislationDiscriminator
MERGE|CalculationCard|VISION|SD_E17062|Statutory Deductions|AU-LDG|2020/06/17|E17062|AU

METADATA|CardComponent|SourceSystemOwner|SourceSystemId|DirCardCompDefName|LegislativeDataGroupName|
EffectiveStartDate|DirCardId(SourceSystemId)
MERGE|CardComponent|VISION|SD_E17062_SAGC|Superannuation Guarantee Contributions|AU-LDG|2020/06/17|SD_E17062
MERGE|CardComponent|VISION|SD_E17062_EESC|Employee Superannuation Contributions|AU-LDG|2020/06/17|SD_E17062
MERGE|CardComponent|VISION|SD_E17062_ERSC|Additional Employer Superannuation Contributions|AU-LDG|
2020/06/17|SD_E17062
MERGE|CardComponent|VISION|SD_E17062_PAYG|PAYG Withholding|AU-LDG|2020/06/17|SD_E17062
MERGE|CardComponent|VISION|SD_E17062_PYTX|Payroll Tax|AU-LDG|2020/06/17|SD_E17062

METADATA|ComponentDetail|SourceSystemOwner|SourceSystemId|DirCardCompDefName|
LegislativeDataGroupName|EffectiveStartDate|DirCardCompId(SourceSystemId)|LegislationDiscriminator|
FLEX:Deduction Developer DF|DirInformationCategory|_SUPER_FUND_ID_Display(Deduction
 Developer DF=HRX_AU_SG_CONTRIBUTIONS)|_DATE_VALID_CHOICE_ACCEPTED(Deduction Developer
 DF=HRX_AU_SG_CONTRIBUTIONS)|_DATE_VALID_CHOICE_COMMENCED(Deduction Developer
 DF=HRX_AU_SG_CONTRIBUTIONS)|_MEMBER_NUMBER(Deduction Developer DF=HRX_AU_SG_CONTRIBUTIONS)|
_OVRD_MAX_CONRIBUTION_BASE_FLAG(Deduction Developer DF=HRX_AU_SG_CONTRIBUTIONS)|
_OVRD_MIN_HOURS_FLAG(Deduction Developer DF=HRX_AU_SG_CONTRIBUTIONS)|_OVRD_MIN_MAX_AGE_FLAG(Deduction
```

```
 Developer DF=HRX_AU_SG_CONTRIBUTIONS)|_OVRD_MONTHLY_MIN_EARNINGS_FLAG(Deduction Developer
 DF=HRX_AU_SG_CONTRIBUTIONS)|_SGC_UNDER_18_HOURS(Deduction Developer DF=HRX_AU_SG_CONTRIBUTIONS)|
_SUPER_FUND_ID_Display(Deduction Developer DF=HRX_AU_PERSONAL_SUPER_CONTRIBUTIONS)|
_MEMBER_NUMBER(Deduction Developer DF=HRX_AU_PERSONAL_SUPER_CONTRIBUTIONS)|_PRETAX_FLAG(Deduction
 Developer DF=HRX_AU_PERSONAL_SUPER_CONTRIBUTIONS)|_SUPER_FUND_ID_Display(Deduction
 Developer DF=HRX_AU_ADDL_EMPLOYER_SUPER_CONTRIBUTIONS)|_MEMBER_NUMBER(Deduction
 Developer DF=HRX_AU_ADDL_EMPLOYER_SUPER_CONTRIBUTIONS)|_RESC_FLAG(Deduction Developer
 DF=HRX_AU_ADDL_EMPLOYER_SUPER_CONTRIBUTIONS)|contributionType(Deduction Developer
 DF=HRX_AU_ADDL_EMPLOYER_SUPER_CONTRIBUTIONS)|_AU_RESIDENT_FLAG(Deduction Developer
 DF=HRX_AU_TFN_DECLARATION)|_DECLARATION_DATE(Deduction Developer DF=HRX_AU_TFN_DECLARATION)|stsl(Deduction
 Developer DF=HRX_AU_TFN_DECLARATION)|_PAYMENT_BASIS(Deduction Developer DF=HRX_AU_TFN_DECLARATION)|
_SENIOR_AU_TYPE(Deduction Developer DF=HRX_AU_TFN_DECLARATION)|_SUBMISSION_DATE_TO_ATO(Deduction Developer
 DF=HRX_AU_TFN_DECLARATION)|_TAX_FREE_THRESHOLD_FLAG(Deduction Developer DF=HRX_AU_TFN_DECLARATION)|
_TAX_OFFSET_AMOUNT(Deduction Developer DF=HRX_AU_TFN_DECLARATION)|_TFN(Deduction Developer
 DF=HRX_AU_TFN_DECLARATION)|whmCountryCode_Display(Deduction Developer DF=HRX_AU_TFN_DECLARATION)|
workingHolidayMaker(Deduction Developer DF=HRX_AU_TFN_DECLARATION)|_INCLUDE_BONUS_FLAG(Deduction
 Developer DF=HRX_AU_PAYGW_VARIATION)|_VARIATION_TYPE(Deduction Developer DF=HRX_AU_PAYGW_VARIATION)|
_EXEMPTION_TYPE(Deduction Developer DF=HRX_AU_MEDICARE_LEVY_VARIATION)|_INCOME_LT_THRESHOLD_FLAG(Deduction
 Developer DF=HRX_AU_MEDICARE_LEVY_VARIATION)|_NUMBER_OF_DEPENDENT_CHILDREN(Deduction Developer
 DF=HRX_AU_MEDICARE_LEVY_VARIATION)|_SPOUSE_FLAG(Deduction Developer DF=HRX_AU_MEDICARE_LEVY_VARIATION)|
_SURCHARGE_FLAG(Deduction Developer DF=HRX_AU_MEDICARE_LEVY_VARIATION)|_EXCLUDE_FLAG(Deduction Developer
 DF=HRX_AU_PAYROLL_TAX)|_STATE_Display(Deduction Developer DF=HRX_AU_PAYROLL_TAX)
MERGE|ComponentDetail|VISION|SD_E17062_SAGC|Superannuation Guarantee Contributions|AU-LDG|2020/06/17|
SD_E17062_SAGC|AU|HRX_AU_SG_CONTRIBUTIONS|HRX_AU_SG_CONTRIBUTIONS|AU-Super Fund|2020/06/17|2020/06/17|
22JUN202301|Y|Y|Y|Y|7|||||||||||||||||||||||||||
MERGE|ComponentDetail|VISION|SD_E17062_EESC|Employee Superannuation Contributions|AU-LDG|2020/06/17|
SD_E17062_EESC|AU|HRX_AU_PERSONAL_SUPER_CONTRIBUTIONS|HRX_AU_PERSONAL_SUPER_CONTRIBUTIONS|||||||||AU-Super
 Fund|123456|Y|||||||||||||||||||||||
MERGE|ComponentDetail|VISION|SD_E17062_ERSC|Additional Employer Superannuation
 Contributions|AU-LDG|2020/06/17|SD_E17062_ERSC|AU|HRX_AU_ADDL_EMPLOYER_SUPER_CONTRIBUTIONS|
HRX_AU_ADDL_EMPLOYER_SUPER_CONTRIBUTIONS|||||||||||||AU-Super Fund|231231001|Y|
ORA_VOLUNTARY|||||||||||||||||||
MERGE|ComponentDetail|VISION|SD_E17062_PAYG_TFN|PAYG Withholding|AU-LDG|2020/06/17|SD_E17062_PAYG|AU|
HRX_AU_TFN_DECLARATION|HRX_AU_TFN_DECLARATION|||||||||||||||||N|2020/06/17|N|FULL|SINGLE|2020/06/17|N|50|
110987566|India|Y|||||||||
MERGE|ComponentDetail|VISION|SD_E17062_PAYG_GWV|PAYG Withholding|AU-LDG|2020/06/17|SD_E17062_PAYG|AU|
HRX_AU_PAYGW_VARIATION|HRX_AU_PAYGW_VARIATION|||||||||||||||||||||||||||||N|INCREASE|||||||
MERGE|ComponentDetail|VISION|SD_E17062_PAYG_MED|PAYG Withholding|AU-LDG|2020/06/17|SD_E17062_PAYG|AU|
HRX_AU_MEDICARE_LEVY_VARIATION|HRX_AU_MEDICARE_LEVY_VARIATION|||||||||||||||||||||||||||||FULL|N|2|N|
0.0125||
MERGE|ComponentDetail|VISION|SD_E17062_PYTX|Payroll Tax|AU-LDG|2020/06/17|SD_E17062_PYTX|AU|
HRX_AU_PAYROLL_TAX|HRX_AU_PAYROLL_TAX|||||||||||||||||||||||||||||||||Y|VIC

METADATA|CardAssociation|SourceSystemOwner|SourceSystemId|DirCardId(SourceSystemId)|
LegislativeDataGroupName|EffectiveStartDate|TaxReportingUnitName
MERGE|CardAssociation|VISION|SD_E17062_AS_AU_TRU|SD_E17062|AU-LDG|2020/06/17|AU TRU

METADATA|CardAssociationDetail|SourceSystemOwner|SourceSystemId|DirRepCardId(SourceSystemId)|
AssignmentNumber|EffectiveStartDate
MERGE|CardAssociationDetail|VISION|SD_E17062_AD_AU_TRU|SD_E17062_AS_AU_TRU|E17062|2020/06/17
```

34 Loading Payroll Localization Data for India

Initializing Balances

Overview of Balance Initialization for India

In addition to the general guidance provided in the Overview of Balance Initialization topic, consider the following when initializing balances for India.

- For the Professional Tax balances, you need to set the **Area1** context.

    a. To set the **Area1** context for the dimensions **Core Relationship Area1 Tax Year to Date** and **Core Relationship Area1 Half Year to Date**, pass the state code context under **AreaOne**.
    b. Derive the corresponding **AreaOne** value (lookup_code) for a particular state from the lookup **ORA_HRX_IN_PT_STATES**. For more details on the area code for professional tax, see *Balances to Initialize for India*.

- For social insurance balances, you need to set the **Organization** context, **hr_all_organization_units_vl.organization_id**.

    Set the Organization context for the dimension **Relationship Organization Tax Year to Date**, by passing the **ContextOneName** as **Organization ID** and the corresponding Organization_ID from **hr_all_organization_units_vl** as the **ContextOneValue** value, for the below social insurance balances:

    - Employer LWF Contribution
    - Employee LWF Contribution
    - Employer NPS Contribution
    - Employee NPS Tier 1 Contribution
    - Employee NPS Tier 2 Contribution
    - NPS Computation Salary
    - Employee PF Contribution
    - Employee Voluntary PF Contribution
    - Employee PF Contribution Arrears
    - Employee Voluntary PF Contribution Arrears
    - Employer PF Contribution

- For earnings, tax related and **Taxable House Rent Allowance** balances, the **Tax Unit** context needs to be set. Set the **TaxUnitName** context to initialize the dimension **Relationship Tax Unit Tax Year to Date** for the following balances:
  - Earnings
  - Income Tax This Pay
  - Surcharge This Pay
  - Health and Education Cess This Pay
  - Taxable House Rent Allowance
- For initializing the Professional Tax (PT) balances, here are the points to note:
-

Deduction Frequency in State	Salary Projection	Considerations
Monthly	Monthly	The Period to Date(PTD) value of PT computational salary for that state, is used t decide the PT slabs and deduction amount. For example: Nagaland The professional tax amount already deducted is considered for **Deductions u/s 16**.
Monthly	Annual	The Tax Year to Date(TYTD) value of PT computational salary for that state is used to decide the PT slabs and deduction amount. For example: Odisha The professional tax amount already deducted will be considered for **Deduction u/s 16** and PT computational salary value for that state, is used to project the annual salary for PT calculation.
Annual	Annual	The TYTD value of PT computational salary for that state is used to decide the P slabs and deduction amount. The professional tax amount already deducted is considered for **Deductions u/s 16** and PT computational salary value for that state, is used to project the annual salary for PT calculation. For example: For Bihar, PT is deducted in the September payroll run. If payroll is being run from October onward, then initialize professional tax (amount which has been already deducted) for **Deductions u/s 16** and PT computational salary value for Bihar to project the annual salary for PT calculation. This is because PT will not be deducted after September for Bihar. If payroll is set to be run in September or prior, then we can initialize the PT computational salary for that state along with professional tax as 0. In case of change is PT slabs in current system, exemption in **Deductions u/s 16** will be provided as per the slab in current payroll. Any other PT that needs to be deducted is adjusted manually.
Half Yearly	Half Yearly	The THYTD value of PT computational salary for a state is used to decide the PT slabs and deduction amount.

Deduction Frequency in State	Salary Projection	Considerations
		Ideally, we expect the deduction month at TRU to be kept same in both legacy and in Fusion. For example, if the deduction month is set to June for the first half year, and PT has already been deducted in the legacy system and the PT amount initialized as of 01-June, then when the payroll runs for June, the PT amount will be deducted in current system. This needs to be adjusted manually.
Monthly	Half Yearly	The THYTD value of PT computational salary for a state is used to calculate the PT slabs and deduction amount. If the first payroll run in Fusion is in June, PT will be deducted as per the slab the employee falls in the month of June. Any change needs to be done manually. Initialize the below dimensions in case of Half-Yearly projections: **Professional Tax Core Relationship Area1 Half Year to Date** **PT Computational Salary for <State> Relationship Tax Half Year to Date**

- For the ESI balances, the **ESI Eligible Salary** is used to decide the eligibility of the employee. Based on the month from which ESI eligibility needs to be checked, the corresponding month balance can be initialized. For example, if the first payroll run is in August, and the employee eligibility needs to be decide based on April salary, then initialize the value for the dimension: **Relationship Last April 1 Month**.

  Since ESI is deducted every month, we don't account for what was already deducted in legacy system. We expect it to be deducted correctly and reported. We're only initializing the eligibility salary.
- For calculating the Gratuity balances, both the government sector employees and private sector, who are covered by the Gratuity Act are considered. As the last month's salary is used for gratuity calculation, the dimension **Relationship Last Month** can be initialized for the balance **Gratuity Computation Salary**.

  For private sector employees not covered by the Gratuity Act, the last 10 months salary is used for gratuity calculation. Hence the balance value **Gratuity Computation Salary** for the last 10 months can be initialized. For example: Relationship Last July 1 Month, Relationship Last June 1 Month and so on.
- For the leave encashment balance **Leave Encashment Computation Salary**, if the employee's termination date falls on the last day of the month, then initialize the dimension for the previous 9 months, and calculate the value using the current month's salary and previous 9 months' salary.

  If the employee's termination date doesn't fall on the last day of the month, then initialize the dimension for the previous 10 months, and calculate the value using the previous 10 months' salary.
- For the retrenchment compensation balance **Retrenchment Compensation Computation Salary**, if the employees' termination date falls on the last day of the month, then preceding two months dimension needs to be initialized. If the employees termination date doesn't fall on the last day of the month, then preceding three months dimension needs to be initialized.

- For calculating the Commuted Pension balance, if gratuity amount needs to be considered then initialize the **Gratuity Amount** balance. Similarly, the correct month's dimension needs to be initialized, if you run payroll in the second half of the financial year.

  > **Note:** If you initialized any of the below balances with a value, then you need to initialize the **Earnings** balance with the same amount. The salary in section 17 reflects the termination payment amount.
  >
  > - Gratuity Amount
  > - Leave Encashment
  > - Retrenchment Compensation Amount
  > - Commuted Pension
  > - VRS Amount

- For HRA, the below balances need to be initialized, with each month's correct value:
  - Salary for House Rent Allowance Exemption
  - Taxable House Rent Allowance
  - House Rent Allowance Balance
  - For **Taxable House Rent Allowance** ensure that the value in **Relationship Tax Unit Tax Year to Date** matches the sum of the individual months value.

    Example. If Relationship Tax Unit Tax Year to Date is 2400, then

    Relationship Last April 1 Month -- 600

    Relationship Last May 1 Month -- 600

    Relationship Last June 1 Month -- 600

    Relationship Last July 1 Month -- 600 is a valid combination.

Related Topics

- Overview of Balance Initialization
- Steps to Initialize Balances
- Balances to Initialize for India

Balances to Initialize for India

The India Legislative Balances are balances that Oracle Payroll Cloud uses to perform tax reporting and payroll tax calculations.

Balance initialization is required for the following categories of balances for India:

- Earnings balances
- Allowance balances
- Social Insurance balances
- Termination and Superannuation
- HRA related balances

Earnings Balances

Balance Name	Dimension Name
Earnings	Relationship Tax Unit Tax Year to Date
Income Tax This Pay	
Surcharge This Pay	
Health and Education Cess This Pay	
Taxable Perquisites	Relationship Tax Year to Date

Allowance Balances

Balance Name	Dimension Name
Scholarship Allowance Nontaxable	Relationship Tax Year to Date
Academic Allowance Nontaxable	
Academic Allowance Taxable	
Conveyance Allowance Nontaxable	
Conveyance Allowance Taxable	
Compensatory Field Area Allowance Nontaxable	
Compensatory Field Area Allowance Taxable	

Balance Name	Dimension Name
Compensatory Modified Field Area Allowance Nontaxable	
Compensatory Modified Field Area Allowance Taxable	
Daily Allowance Nontaxable	
Daily Allowance Taxable	
Helper Allowance Nontaxable	
Helper Allowance Taxable	
Travel Allowance Nontaxable	
Travel Allowance Taxable	
Tribal Area Allowance Nontaxable	Relationship Tax Year to Date
Tribal Area Allowance Taxable	
Transport Allowance for Physically Disabled Nontaxable	
Transport Allowance for Physically Disabled Taxable	
Uniform Allowance Nontaxable	
Uniform Allowance Taxable	
Under Ground Allowance Nontaxable	
Under Ground Allowance Taxable	
High Altitude Allowance Nontaxable	
High Altitude Allowance Taxable	
Special Compensatory Allowance Nontaxable	

Balance Name	Dimension Name
Special Compensatory Allowance Taxable	
Children Education Allowance Nontaxable	
Children Education Allowance Taxable	
Hostel Expenditure Allowance Nontaxable	
Hostel Expenditure Allowance Taxable	

Social Insurance Balances

Balance Name	Dimension Name
Employer LWF Contribution	Relationship Organization Tax Year to Date
Employee LWF Contribution	
Employer NPS Contribution	
Employee NPS Tier 1 Contribution	
Employee NPS Tier 2 Contribution	
NPS Computation Salary	
Employee PF Contribution	
Employee Voluntary PF Contribution	
Employee PF Contribution Arrears	
Employee Voluntary PF Contribution Arrears	
Employer PF Contribution	

Employee State Insurance Balances

Balance Name	Dimension Name
ESI Eligible Salary	Relationship Last April 1 Month
	Relationship Last May 1 Month
	Relationship Last June 1 Month
	Relationship Last July 1 Month
	Relationship Last August 1 Month
	Relationship Last September 1 Month
	Relationship Last October 1 Month
	Relationship Last November 1 Month
	Relationship Last December 1 Month
	Relationship Last January 1 Month
	Relationship Last February 1 Month
	Relationship Last March 1 Month

Professional Tax Balances

Balance Name	Dimension Name
Professional Tax	Core Relationship Area1 Tax Year to Date
	Core Relationship Area1 Half Year to Date
PT Computational Salary for Andhra Pradesh	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Assam	Relationship Tax Year to Date
	Relationship Tax Half Year to Date

Balance Name	Dimension Name
PT Computational Salary for Bihar	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Gujarat	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Jharkhand	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Karnataka	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Kerala	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Maharashtra	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Meghalaya	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Manipur	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Madhya Pradesh	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Mizoram	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Nagaland	Relationship Tax Year to Date

Balance Name	Dimension Name
	Relationship Tax Half Year to Date
PT Computational Salary for Odisha	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Punjab	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Puducherry	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Sikkim	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Chennai corporation	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Coimbatore corporation	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Dindigul corporation	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Madurai corporation	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Panchayats	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Salem corporation	Relationship Tax Year to Date
	Relationship Tax Half Year to Date

Balance Name	Dimension Name
PT Computational Salary for Tirunelveli corporation	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Tripura	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for Telangana	Relationship Tax Year to Date
	Relationship Tax Half Year to Date
PT Computational Salary for West Bengal	Relationship Tax Year to Date
	Relationship Tax Half Year to Date

Termination and Superannuation Balances

Balance Name	Dimension Name
Gratuity Computation Salary	Relationship Last April 1 Month
	Relationship Last May 1 Month
	Relationship Last June 1 Month
	Relationship Last July 1 Month
	Relationship Last August 1 Month
	Relationship Last September 1 Month
	Relationship Last October 1 Month
	Relationship Last November 1 Month
	Relationship Last December 1 Month
	Relationship Last January 1 Month

Balance Name	Dimension Name
	Relationship Last February 1 Month
	Relationship Last March 1 Month
	Relationship Last Month
Gratuity Amount	Relationship Tax Year to Date
Gratuity Amount Nontaxable	Relationship Tax Year to Date
Leave Encashment Computation Salary	Relationship Last Month
	Relationship Last April 1 Month
	Relationship Last May 1 Month
	Relationship Last June 1 Month
	Relationship Last July 1 Month
	Relationship Last August 1 Month
	Relationship Last September 1 Month
	Relationship Last October 1 Month
	Relationship Last November 1 Month
	Relationship Last December 1 Month
	Relationship Last January 1 Month
	Relationship Last February 1 Month
	Relationship Last March 1 Month
Leave Encashment	Relationship Tax Year to Date
Leave Encashment Amount Nontaxable	Relationship Tax Year to Date
Retrenchment Compensation Computation Salary	Relationship Last April 1 Month

Balance Name	Dimension Name
	Relationship Last May 1 Month
	Relationship Last June 1 Month
	Relationship Last July 1 Month
	Relationship Last August 1 Month
	Relationship Last September 1 Month
	Relationship Last October 1 Month
	Relationship Last November 1 Month
	Relationship Last December 1 Month
	Relationship Last January 1 Month
	Relationship Last February 1 Month
	Relationship Last March 1 Month
Retrenchment Compensation Nontaxable	Relationship Tax Year to Date
Retrenchment Compensation Amount	Relationship Tax Year to Date
Commuted Pension	Relationship Tax Year to Date
Commuted Pension Nontaxable	Relationship Tax Year to Date
VRS Amount	Relationship Tax Year to Date
VRS Amount Nontaxable	Relationship Tax Year to Date
Superannuation Contribution	Relationship Tax Year to Date

HRA Related Balances

Taxable House Rent Allowance	Relationship Tax Unit Tax Year to Date

	Relationship Last April 1 Month
	Relationship Last May 1 Month
	Relationship Last June 1 Month
	Relationship Last July 1 Month
	Relationship Last August 1 Month
	Relationship Last September 1 Month
	Relationship Last October 1 Month
	Relationship Last November 1 Month
	Relationship Last December 1 Month
	Relationship Last January 1 Month
	Relationship Last February 1 Month
	Relationship Last March 1 Month
House Rent Allowance Balance	Relationship Last April 1 Month
	Relationship Last May 1 Month
	Relationship Last June 1 Month
	Relationship Last July 1 Month
	Relationship Last August 1 Month
	Relationship Last September 1 Month
	Relationship Last October 1 Month
	Relationship Last November 1 Month
	Relationship Last December 1 Month

	Relationship Last January 1 Month
	Relationship Last February 1 Month
	Relationship Last March 1 Month
Salary for House Rent Allowance Exemption	Relationship Last April 1 Month
	Relationship Last May 1 Month
	Relationship Last June 1 Month
	Relationship Last July 1 Month
	Relationship Last August 1 Month
	Relationship Last September 1 Month
	Relationship Last October 1 Month
	Relationship Last November 1 Month
	Relationship Last December 1 Month
	Relationship Last January 1 Month
	Relationship Last February 1 Month
	Relationship Last March 1 Month

Area Code for Professional Tax

State	AreaOne
Andhra Pradesh	10001
Telangana	10002
Karnataka	10003

State	AreaOne
Odisha	10004
Assam	10005
Gujarat	10006
West Bengal	10007
Madhya Pradesh	10008
Maharashtra	10009
Jharkhand	10010
Punjab	10011
Kerala	10012
Chennai	10013
Bihar	10014
Salem	10015
Coimbatore	10016
Tirunelveli	10017
Madurai	10018
Dindigul	10019
Panchayats in Tamil Nadu	10020
Tripura	10021
Manipur	10022
Meghalaya	10023
Mizoram	10024

State	AreaOne
Nagaland	10025
Sikkim	10026
Puducherry	10027

Example of Loading Earnings Balances for India

In this example, you create balance initialization records to initialize earnings balances for India.

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|BALANCE_INIT_EARNINGS|2023/08/01|ZHRX_IN_LDG
```

Load the Initialize Balance Batch Lines

In the following example, several balances are being initialized for the same employee. The employee is identified by their payroll relationship number, employment terms number and assignment number.

Attribute	Value
PayrollRelationshipNumber	966169008890480
TermNumber	ET966169008890480
AssignmentNumber	E966169008890480

The employee's payroll and tax reporting unit must also be specified on every initialize balance line.

Attribute	Value
PayrollName	IN MONTHLY PAYROLL 2021
TaxUnitName	ZHRX_IN_TAX_TRU

Load these details to initialize the Earnings balance to 6000:

Attribute	Value
LineSequence	1

Attribute	Value
BalanceName	Earnings
DimensionName	Relationship Tax Unit Tax Year to Date
Value	6000
ContextOneName	
ContextOneValue	
AreaOne	

Load these details to initialize the Income Tax This Pay balance to 4000:

Attribute	Value
LineSequence	2
BalanceName	Income Tax This Pay
DimensionName	Relationship Tax Unit Tax Year to Date
Value	4000
ContextOneName	
ContextOneValue	
AreaOne	

Load these details to initialize the Surcharge This Pay balance to 2000:

Attribute	Value
LineSequence	3
BalanceName	Surcharge This Pay
DimensionName	Relationship Tax Unit Tax Year to Date
Value	2000
ContextOneName	
ContextOneValue	
AreaOne	

Load these details to initialize the Health and Education Cess This Pay balance:

Attribute	Value
LineSequence	4

Attribute	Value
BalanceName	Health and Education Cess This Pay
DimensionName	Relationship Tax Unit Tax Year to Date
Value	800
ContextOneName	
ContextOneValue	
AreaOne	

Load these details to initialize the Taxable Perquisites balance:

Attribute	Value
LineSequence	5
BalanceName	Taxable Perquisites
DimensionName	Relationship Tax Year to Date
Value	6001
ContextOneName	
ContextOneValue	
AreaOne	

Use the InitializeBalanceBatchLine.dat file to load the balance initialization lines defined above:

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
Value|ContextOneName|ContextOneValue|AreaOne
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_EARNINGS|1|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021|ZHRX_IN_TAX_TRU|Earnings|Relationship Tax Unit Tax Year to Date|
6000|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_EARNINGS|2|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021|ZHRX_IN_TAX_TRU|Income Tax This Pay|Relationship Tax Unit Tax Year
 to Date|4000|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_EARNINGS|3|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021|ZHRX_IN_TAX_TRU|Surcharge This Pay|Relationship Tax Unit Tax Year
 to Date|2000|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_EARNINGS|4|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021|ZHRX_IN_TAX_TRU|Health and Education Cess This Pay|Relationship Tax
 Unit Tax Year to Date|800|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_EARNINGS|5|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Taxable Perquisites|Relationship Tax Year to Date|6001|||
```

Related Topics

- Overview of Balance Initialization
- Steps to Initialize Balances
- Overview of Balance Initialization for India

Example of Loading Allowance Balances for India

In this example, you create balance initialization records to initialize allowance balances for India.

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|BALANCE_INIT_ALLOWANCE|2023/08/01|ZHRX_IN_LDG
```

Load the Initialize Balance Batch Lines

In the following example, several balances are being initialized for the same employee. The employee is identified by their payroll relationship number, employment terms number and assignment number.

Attribute	Value
PayrollRelationshipNumber	955160008194736
TermNumber	ET955160008194736
AssignmentNumber	E955160008194736

The employee's payroll and tax reporting unit must also be specified on every initialize balance line.

Attribute	Value
PayrollName	IN MONTHLY PAYROLL 2021
TaxUnitName	

Load these details to initialize the Scholarship Allowance Nontaxable balance to:

Attribute	Value
LineSequence	1
BalanceName	Scholarship Allowance Nontaxable
DimensionName	Relationship Tax Year to Date
Value	2001
ContextOneName	
ContextOneValue	
AreaOne	

Load these details to initialize the Academic Allowance Nontaxable balance to:

Attribute	Value
LineSequence	2
BalanceName	Academic Allowance Nontaxable
DimensionName	Relationship Tax Year to Date
Value	2002
ContextOneName	
ContextOneValue	
AreaOne	

Load these details to initialize the Academic Allowance Taxable balance to:

Attribute	Value
LineSequence	3
BalanceName	Academic Allowance Taxable
DimensionName	Relationship Tax Year to Date
Value	2003
ContextOneName	
ContextOneValue	
AreaOne	

Load these details to initialize the Conveyance Allowance Nontaxable balance:

Attribute	Value
LineSequence	4
BalanceName	Conveyance Allowance Nontaxable
DimensionName	Relationship Tax Year to Date
Value	2004
ContextOneName	
ContextOneValue	
AreaOne	

Load these details to initialize the Conveyance Allowance Taxable balance:

Attribute	Value
LineSequence	5
BalanceName	Conveyance Allowance Taxable
DimensionName	Relationship Tax Year to Date
Value	2005
ContextOneName	
ContextOneValue	
AreaOne	

Load these details to initialize the Compensatory Field Area Allowance Nontaxable balance to:

Attribute	Value
LineSequence	6
BalanceName	Compensatory Field Area Allowance Nontaxable
DimensionName	Relationship Tax Year to Date
Value	2006
ContextOneName	
ContextOneValue	
AreaOne	

Load these details to initialize the Compensatory Field Area Allowance Taxable balance to:

Attribute	Value
LineSequence	7
BalanceName	Compensatory Field Area Allowance Taxable
DimensionName	Relationship Tax Year to Date
Value	2007
ContextOneName	
ContextOneValue	
AreaOne	

Use the InitializeBalanceBatchLine.dat file to load the balance initialization lines defined above:

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
Value|ContextOneName|ContextOneValue|AreaOne
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_ALLOWANCE|1|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Scholarship Allowance Nontaxable|Relationship Tax Year to Date|
2001|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_ALLOWANCE|2|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Academic Allowance Nontaxable|Relationship Tax Year to Date|
2002|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_ALLOWANCE|3|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Academic Allowance Taxable|Relationship Tax Year to Date|2003|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_ALLOWANCE|4|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Conveyance Allowance Nontaxable|Relationship Tax Year to Date|
2004|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_ALLOWANCE|5|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Conveyance Allowance Taxable|Relationship Tax Year to Date|2005|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_ALLOWANCE|6|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Compensatory Field Area Allowance Nontaxable|Relationship Tax Year
 to Date|2006|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_ALLOWANCE|7|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Compensatory Field Area Allowance Taxable|Relationship Tax Year to
 Date|2007|||
```

Related Topics

- Overview of Balance Initialization
- Steps to Initialize Balances
- Overview of Balance Initialization for India

Example of Loading Social Insurance Balances for India

In this example, you create balance initialization records to initialize social insurance balances for India.

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|BALANCE_INIT_SOCIAL_INS|2023/08/01|ZHRX_IN_LDG
```

Load the Initialize Balance Batch Lines

In the following example, several balances are being initialized for the same employee. The employee is identified by their payroll relationship number, employment terms number and assignment number.

Attribute	Value
PayrollRelationshipNumber	966169008890480
TermNumber	ET966169008890480
AssignmentNumber	E966169008890480

The employee's payroll and tax reporting unit must also be specified on every initialize balance line.

Attribute	Value
PayrollName	IN MONTHLY PAYROLL 2021
TaxUnitName	

Load these details to initialize the Employer LWF Contribution balance to:

Attribute	Value
LineSequence	1
BalanceName	Employer LWF Contribution
DimensionName	Relationship Organization Tax Year to Date
Value	40
ContextOneName	Organization ID
ContextOneValue	300100564984701

Load these details to initialize the Employee LWF Contribution balance to:

Attribute	Value
LineSequence	2
BalanceName	Employee LWF Contribution
DimensionName	Relationship Organization Tax Year to Date
Value	20
ContextOneName	Organization ID|
ContextOneValue	300100564984701

Load these details to initialize the Employer NPS Contribution balance to:

Attribute	Value
LineSequence	3
BalanceName	Employer NPS Contribution
DimensionName	Relationship Organization Tax Year to Date
Value	1800
ContextOneName	Organization ID
ContextOneValue	300100590461999

Load these details to initialize the Employee NPS Tier 1 Contribution balance to:

Attribute	Value
LineSequence	4
BalanceName	Employee NPS Tier 1 Contribution
DimensionName	Relationship Organization Tax Year to Date
Value	2200
ContextOneName	Organization ID
ContextOneValue	300100590461999

Load these details to initialize the Employee NPS Tier 2 Contribution balance to:

Attribute	Value
LineSequence	5
BalanceName	Employee NPS Tier 2 Contribution
DimensionName	Relationship Organization Tax Year to Date
Value	1600
ContextOneName	Organization ID
ContextOneValue	300100590461999

Load these details to initialize the NPS Computation Salary balance to:

Attribute	Value
LineSequence	6
BalanceName	NPS Computation Salary
DimensionName	Relationship Organization Tax Year to Date
Value	30000
ContextOneName	Organization ID
ContextOneValue	300100590461999

Load these details to initialize the ESI Eligible Salary balance to:

Attribute	Value
LineSequence	7

Attribute	Value
BalanceName	ESI Eligible Salary
DimensionName	Relationship Last April 1 Month
Value	18000
ContextOneName	
ContextOneValue	

Load these details to initialize the Employee PF Contribution balance to:

Attribute	Value
LineSequence	8
BalanceName	Employee PF Contribution
DimensionName	Relationship Organization Tax Year to Date
Value	5000
ContextOneName	Organization ID
ContextOneValue	300100552708841

Load these details to initialize the Employee Voluntary PF Contribution balance to:

Attribute	Value
LineSequence	9
BalanceName	Employee Voluntary PF Contribution
DimensionName	Relationship Organization Tax Year to Date
Value	6000
ContextOneName	Organization ID
ContextOneValue	300100552708841

Load these details to initialize the Employee PF Contribution Arrears balance to:

Attribute	Value
LineSequence	10
BalanceName	Employee PF Contribution Arrears
DimensionName	Relationship Organization Tax Year to Date
Value	7000

Attribute	Value
ContextOneName	Organization ID
ContextOneValue	300100552708841

Load these details to initialize the Employee Voluntary PF Contribution Arrears balance to:

Attribute	Value
LineSequence	11
BalanceName	Employee Voluntary PF Contribution Arrears
DimensionName	Relationship Organization Tax Year to Date
Value	8000
ContextOneName	Organization ID
ContextOneValue	300100552708841

Load these details to initialize the Employer PF Contribution balance to:

Attribute	Value
LineSequence	12
BalanceName	Employer PF Contribution
DimensionName	Relationship Organization Tax Year to Date
Value	888000
ContextOneName	Organization ID
ContextOneValue	300100552708841

Load these details to initialize the Professional Tax balance to:

Attribute	Value
LineSequence	13
BalanceName	Professional Tax
DimensionName	Core Relationship Area1 Tax Year to Date
Value	300
ContextOneName	
ContextOneValue	
AreaOne	10027

Load these details to initialize the Professional Tax balance to:

Attribute	Value
LineSequence	14
BalanceName	Professional Tax
DimensionName	Core Relationship Area1 Half Year to Date
Value	300
ContextOneName	
ContextOneValue	10027

Load these details to initialize the PT Computational Salary for Puducherry balance to:

Attribute	Value
LineSequence	15
BalanceName	PT Computational Salary for Puducherry
DimensionName	Relationship Tax Year to Date
Value	120000
ContextOneName	
ContextOneValue	

Load these details to initialize the PT Computational Salary for Puducherry balance to:

Attribute	Value
LineSequence	16
BalanceName	PT Computational Salary for Puducherry
DimensionName	Relationship Tax Half Year to Date
Value	120000
ContextOneName	
ContextOneValue	

Use the InitializeBalanceBatchLine.dat file to load the balance initialization lines defined above:

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
Value|ContextOneName|ContextOneValue|AreaOne
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|1|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Employer LWF Contribution|Relationship Organization Tax Year to
 Date|40|Organization ID|300100564984701|
```

```
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|2|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Employee LWF Contribution|Relationship Organization Tax Year to
 Date|20|Organization ID|300100564984701|
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|3|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Employer NPS Contribution|Relationship Organization Tax Year to
 Date|1800|Organization ID|300100590461999|
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|4|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Employee NPS Tier 1 Contribution|Relationship Organization Tax
 Year to Date|2200|Organization ID|300100590461999|
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|5|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Employee NPS Tier 2 Contribution|Relationship Organization Tax
 Year to Date|1600|Organization ID|300100590461999|
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|6|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||NPS Computation Salary|Relationship Organization Tax Year to Date|
30000|Organization ID|300100590461999|
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|7|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||ESI Eligible Salary|Relationship Last April 1 Month|18000|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|8|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Employee PF Contribution|Relationship Organization Tax Year to
 Date|5000|Organization ID|300100552708841|
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|9|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Employee Voluntary PF Contribution|Relationship Organization Tax
 Year to Date|6000|Organization ID|300100552708841|
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|10|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Employee PF Contribution Arrears|Relationship Organization Tax
 Year to Date|7000|Organization ID|300100552708841|
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|11|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Employee Voluntary PF Contribution Arrears|Relationship
 Organization Tax Year to Date|8000|Organization ID|300100552708841|
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|12|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Employer PF Contribution|Relationship Organization Tax Year to
 Date|888000|Organization ID|300100552708841|
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|13|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Professional Tax|Core Relationship Area1 Tax Year to Date|300|||
10027
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|14|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||Professional Tax|Core Relationship Area1 Half Year to Date|300|||
10027
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|15|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||PT Computational Salary for Puducherry|Relationship Tax Year to
 Date|120000|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_SOCIAL_INS|16|966169008890480|ET966169008890480|
E966169008890480|IN MONTHLY PAYROLL 2021||PT Computational Salary for Puducherry|Relationship Tax Half Year
 to Date|120000|||
```

Example of Loading Termination and Superannuation Balances for India

In this example, you create balance initialization records to initialize termination and superannuation balances for India.

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|BALANCE_INIT_TERMPAY|2023/08/01|ZHRX_IN_LDG
```

Load the Initialize Balance Batch Lines

In the following example, several balances are being initialized for the same employee. The employee is identified by their payroll relationship number, employment terms number and assignment number.

Attribute	Value
PayrollRelationshipNumber	955160008194736
TermNumber	ET955160008194736
AssignmentNumber	E955160008194736

The employee's payroll and tax reporting unit must also be specified on every initialize balance line.

Attribute	Value
PayrollName	IN MONTHLY PAYROLL 2021
TaxUnitName	

Load these details to initialize the Gratuity Computation Salary balance to:

Attribute	Value
LineSequence	1
BalanceName	Gratuity Computation Salary
DimensionName	Relationship Last April 1 Month
Value	7001
ContextOneName	
ContextOneValue	

Load these details to initialize the Gratuity Computation Salary balance to:

Attribute	Value
LineSequence	2
BalanceName	Gratuity Computation Salary
DimensionName	Relationship Last May 1 Month
Value	7002
ContextOneName	
ContextOneValue	

Load these details to initialize the Gratuity Computation Salary balance to:

Attribute	Value
LineSequence	3
BalanceName	Gratuity Computation Salary
DimensionName	Relationship Last June 1 Month
Value	7003
ContextOneName	
ContextOneValue	

Load these details to initialize the Gratuity Computation Salary balance to:

Attribute	Value
LineSequence	4
BalanceName	Gratuity Computation Salary
DimensionName	Relationship Last July 1 Month
Value	7004
ContextOneName	
ContextOneValue	

Load these details to initialize the Gratuity Computation Salary balance to:

Attribute	Value
LineSequence	5
BalanceName	Gratuity Computation Salary
DimensionName	Relationship Last Month
Value	7004
ContextOneName	
ContextOneValue	

Load these details to initialize the Gratuity Amount balance to:

Attribute	Value
LineSequence	6

Attribute	Value
BalanceName	Gratuity Amount
DimensionName	Relationship Tax Year to Date
Value	8001
ContextOneName	
ContextOneValue	

Load these details to initialize the Gratuity Amount Nontaxable balance to:

Attribute	Value
LineSequence	7
BalanceName	Gratuity Amount Nontaxable
DimensionName	Relationship Tax Year to Date
Value	|8001
ContextOneName	
ContextOneValue	

Load these details to initialize the Leave Encashment Computation Salary balance to:

Attribute	Value
LineSequence	8
BalanceName	Leave Encashment Computation Salary
DimensionName	Relationship Last Month
Value	9001
ContextOneName	
ContextOneValue	

Load these details to initialize the Leave Encashment balance to:

Attribute	Value
LineSequence	9
BalanceName	Leave Encashment
DimensionName	Relationship Tax Year to Date
Value	9013

Attribute	Value
ContextOneName	
ContextOneValue	

Load these details to initialize the Leave Encashment Amount Nontaxable balance to:

Attribute	Value
LineSequence	10
BalanceName	Leave Encashment Amount Nontaxable
DimensionName	Relationship Tax Year to Date
Value	9013
ContextOneName	
ContextOneValue	

Load these details to initialize the Retrenchment Compensation Computation Salary balance to:

Attribute	Value
LineSequence	11
BalanceName	Retrenchment Compensation Computation Salary
DimensionName	Relationship Last Month
Value	6002
ContextOneName	
ContextOneValue	

Load these details to initialize the Retrenchment Compensation Nontaxable balance to:

Attribute	Value
LineSequence	12
BalanceName	Retrenchment Compensation Nontaxable
DimensionName	Relationship Tax Year to Date
Value	6013
ContextOneName	
ContextOneValue	

Load these details to initialize the Retrenchment Compensation Amount balance to:

Attribute	Value
LineSequence	13
BalanceName	Retrenchment Compensation Amount
DimensionName	Relationship Tax Year to Date
Value	7000
ContextOneName	
ContextOneValue	

Load these details to initialize the Commuted Pension balance to:

Attribute	Value
LineSequence	14
BalanceName	Commuted Pension
DimensionName	Relationship Tax Year to Date
Value	80001
ContextOneName	
ContextOneValue	

Load these details to initialize the Commuted Pension Nontaxable balance to:

Attribute	Value
LineSequence	15
BalanceName	Commuted Pension Nontaxable
DimensionName	Relationship Tax Year to Date
Value	80001
ContextOneName	
ContextOneValue	

Load these details to initialize the VRS Amount balance to:

Attribute	Value
LineSequence	16

Attribute	Value
BalanceName	VRS Amount
DimensionName	Relationship Tax Year to Date
Value	70001
ContextOneName	
ContextOneValue	

Load these details to initialize the VRS Amount Nontaxable balance to:

Attribute	Value
LineSequence	17
BalanceName	VRS Amount Nontaxable
DimensionName	Relationship Tax Year to Date
Value	70001
ContextOneName	
ContextOneValue	

Load these details to initialize the Superannuation Contribution balance to:

Attribute	Value
LineSequence	18
BalanceName	Superannuation Contribution
DimensionName	Relationship Tax Year to Date
Value	55000
ContextOneName	
ContextOneValue	

Use the InitializeBalanceBatchLine.dat file to load the balance initialization lines defined above:

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
Value|ContextOneName|ContextOneValue|AreaOne
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|1|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Gratuity Computation Salary|Relationship Last April 1 Month|
7001|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|2|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Gratuity Computation Salary|Relationship Last May 1 Month|7002|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|3|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Gratuity Computation Salary|Relationship Last June 1 Month|7003|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|4|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Gratuity Computation Salary|Relationship Last July 1 Month|7004|||
```

```
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|5|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Gratuity Computation Salary|Relationship Last Month|7004|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|6|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Gratuity Amount|Relationship Tax Year to Date|8001|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|7|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Gratuity Amount Nontaxable|Relationship Tax Year to Date|8001|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|8|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Leave Encashment Computation Salary|Relationship Last Month|
9001|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|9|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Leave Encashment|Relationship Tax Year to Date|9013|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|10|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Leave Encashment Amount Nontaxable|Relationship Tax Year to Date|
9013|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|11|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Retrenchment Compensation Computation Salary|Relationship Last
 Month|6001|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|12|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Retrenchment Compensation Nontaxable|Relationship Tax Year to
 Date|6013|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|13|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Retrenchment Compensation Amount|Relationship Tax Year to Date|
7000|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|14|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Commuted Pension|Relationship Tax Year to Date|80001|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|15|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Commuted Pension Nontaxable|Relationship Tax Year to Date|80001|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|16|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||VRS Amount|Relationship Tax Year to Date|70001|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|17|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||VRS Amount Nontaxable|Relationship Tax Year to Date|70001|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_TERMPAY|18|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Superannuation Contribution|Relationship Tax Year to Date|55000|||
```

Example of Loading HRA Related Balances for India

In this example, you create balance initialization records to initialize HRA related balances for India.

Load the Initialize Balance Batch Header

Use the InitializeBalanceBatchHeader.dat file to create the batch header.

```
METADATA|InitializeBalanceBatchHeader|BatchName|UploadDate|LegislativeDataGroupName
MERGE|InitializeBalanceBatchHeader|BALANCE_INIT_HRA|2023/08/01|ZHRX_IN_LDG
```

Load the Initialize Balance Batch Lines

In the following example, several balances are being initialized for the same employee. The employee is identified by their payroll relationship number, employment terms number, and assignment number.

Attribute	Value
PayrollRelationshipNumber	955160008194736
TermNumber	ET955160008194736

Attribute	Value
AssignmentNumber	E955160008194736

The employee's payroll and tax reporting unit must also be specified on every initialize balance line.

Attribute	Value
PayrollName	IN MONTHLY PAYROLL 2021
TaxUnitName	

Load these details to initialize the Salary for House Rent Allowance Exemption balance to:

Attribute	Value
LineSequence	1
BalanceName	Salary for House Rent Allowance Exemption
DimensionName	Relationship Last April 1 Month
Value	30001
ContextOneName	
ContextOneValue	

Load these details to initialize the Salary for House Rent Allowance Exemption balance to:

Attribute	Value
LineSequence	2
BalanceName	Salary for House Rent Allowance Exemption
DimensionName	Relationship Last May 1 Month
Value	30002
ContextOneName	
ContextOneValue	

Load these details to initialize the Salary for House Rent Allowance Exemption balance to:

Attribute	Value
LineSequence	3
BalanceName	Salary for House Rent Allowance Exemption

Attribute	Value
DimensionName	Relationship Last June 1 Month
Value	30002
ContextOneName	
ContextOneValue	

Load these details to initialize the Salary for House Rent Allowance Exemption balance to:

Attribute	Value
LineSequence	4
BalanceName	Salary for House Rent Allowance Exemption
DimensionName	Relationship Last July 1 Month
Value	30002
ContextOneName	
ContextOneValue	

Load these details to initialize the Taxable House Rent Allowance balance to:

Attribute	Value
LineSequence	5
Tax Unit Name	ZHRX_IN_TAX_TRU
BalanceName	Taxable House Rent Allowance
DimensionName	Relationship Tax Unit Tax Year to Date
Value	2400
ContextOneName	
ContextOneValue	

Load these details to initialize the Taxable House Rent Allowance balance to:

Attribute	Value
LineSequence	6
BalanceName	Taxable House Rent Allowance
DimensionName	Relationship Last April 1 Month
Value	600

Attribute	Value
ContextOneName	
ContextOneValue	

Load these details to initialize the Taxable House Rent Allowance balance to:

Attribute	Value
LineSequence	7
BalanceName	Taxable House Rent Allowance
DimensionName	Relationship Last May 1 Month
Value	600
ContextOneName	
ContextOneValue	

Load these details to initialize the Taxable House Rent Allowance balance to:

Attribute	Value
LineSequence	8
BalanceName	Taxable House Rent Allowance
DimensionName	Relationship Last June 1 Month
Value	600
ContextOneName	
ContextOneValue	

Load these details to initialize the Taxable House Rent Allowance balance to:

Attribute	Value
LineSequence	9
BalanceName	Taxable House Rent Allowance
DimensionName	Relationship Last July 1 Month
Value	600
ContextOneName	
ContextOneValue	

Load these details to initialize the House Rent Allowance Balance to:

Attribute	Value
LineSequence	10
BalanceName	House Rent Allowance Balance
DimensionName	Relationship Last April 1 Month
Value	800
ContextOneName	
ContextOneValue	

Load these details to initialize the House Rent Allowance Balance to:

Attribute	Value
LineSequence	11
BalanceName	House Rent Allowance Balance
DimensionName	Relationship Last May 1 Month
Value	800
ContextOneName	
ContextOneValue	

Load these details to initialize the House Rent Allowance balance to:

Attribute	Value
LineSequence	12
BalanceName	House Rent Allowance Balance
DimensionName	Relationship Last June 1 Month
Value	800
ContextOneName	
ContextOneValue	

Load these details to initialize the House Rent Allowance balance to:

Attribute	Value
LineSequence	13

Attribute	Value
BalanceName	House Rent Allowance Balance
DimensionName	Relationship Last July 1 Month
Value	800
ContextOneName	
ContextOneValue	

Use the InitializeBalanceBatchLine.dat file to load the balance initialization lines defined above:

```
METADATA|InitializeBalanceBatchLine|LegislativeDataGroupName|BatchName|LineSequence|
PayrollRelationshipNumber|TermNumber|AssignmentNumber|PayrollName|TaxUnitName|BalanceName|DimensionName|
Value|ContextOneName|ContextOneValue|AreaOne
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|1|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Salary for House Rent Allowance Exemption|Relationship Last April
 1 Month|30001|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|2|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Salary for House Rent Allowance Exemption|Relationship Last May 1
 Month|30002|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|3|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Salary for House Rent Allowance Exemption|Relationship Last June 1
 Month|30002|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|4|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Salary for House Rent Allowance Exemption|Relationship Last July 1
 Month|30002|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|5|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021|ZHRX_IN_TAX_TRU|Taxable House Rent Allowance|Relationship Tax Unit
 Tax Year to Date|2400|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|6|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Taxable House Rent Allowance|Relationship Last April 1 Month|
600|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|7|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Taxable House Rent Allowance|Relationship Last May 1 Month|600|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|8|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Taxable House Rent Allowance|Relationship Last June 1 Month|600|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|9|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||Taxable House Rent Allowance|Relationship Last July 1 Month|600|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|10|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||House Rent Allowance Balance|Relationship Last April 1 Month|
800|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|11|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||House Rent Allowance Balance|Relationship Last May 1 Month|800|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|12|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||House Rent Allowance Balance|Relationship Last June 1 Month|800|||
MERGE|InitializeBalanceBatchLine|ZHRX_IN_LDG|BALANCE_INIT_HRA|13|955160008194736|ET955160008194736|
E955160008194736|IN MONTHLY PAYROLL 2021||House Rent Allowance Balance|Relationship Last July 1 Month|800|||
```

35 Loading Talent Objects

Overview of Loading Goals, Goal Plans, and Goal Plan Sets

You can load Goal, Goal Plan, and Goal Plan Set objects using HCM Data Loader. Most of these objects have dependencies on other objects. This table identifies the other objects that must exist in the target environment.

If you're loading some or all of those objects with goals, goal plans, or goal plan sets, then follow the load order shown in the table.

Object	Object Load Order	Notes
Goal Plan	1. Worker and primary assignment 2. Eligibility profile 3. Organization 4. Goal plan document type 5. Goal plan	• Review periods must exist in the target environment before you upload associated goal plans. • Organizations are required for organization goal plans. • A Goal Plan Document Types component is required when the **IncludeInPerfDoc** attribute is set to **Y**.
Goal Plan Set	1. Goal plan 2. Worker and primary assignment 3. Eligibility profile 4. Goal plan set	• Review periods must exist in the target environment before you upload associated goal plan sets. • Goal Plan Set objects can refer to goal plans.
Goal (library)	1. Content item	Library goals are loaded as content items.
Goal (organization)	1. Worker (line manager) 2. Organization 3. Goal plan 4. Goal	For organization goals: • The line manager is the organization owner. • Goal plans are required.
Goal (template)	1. Worker (human resource specialist) and assignment 2. Goal plan 3. Goal	Goal plans are required.
Goal (worker development)	1. Worker 2. Content item 3. Goal	Worker development goals aren't associated with goal plans.
Goal (worker performance)	1. Worker and assignment 2. Goal plan 3. Content item	Worker performance goals must be associated with goal plans.

Object	Object Load Order	Notes
	4. Goal	

Related Topics

- Guidelines for Loading Goals
- Guidelines for Loading Goal Plans
- Guidelines for Loading Goal Plan Sets
- Overview of Goal Management

Guidelines for Loading Goals

Goals represent the objectives of employees and organizations. This topic describes aspects of the Goal object that you must understand to load goals successfully.

Ensure that there are no pending performance or development goal transactions for specific workers before you make any additional changes by uploading goals using HCM Data Loader.

Goal Types

This table identifies the types of goals that you can load using HCM Data Loader.

Goal Type	Description
Organization goal	Created by a manager for a specific organization.
Template goal	Used in goal plans.
Worker goal	Assigned to a worker by a manager or human resource specialist. Two types of worker goals exist: • Development goals • Performance goals

Note: You can also load library goals using HCM Data Loader (HDL). You load library goals as profile content items. You can refer to library goals from other goal objects.

When you load goals in the Goal.dat file, you can include the **GoalTypeCode** attribute. This value is validated using the HRG_GOAL_TYPE lookup type. For organization goals, specify the value of the **GoalTypeCode** attribute as **PERFORMANCE**, which is the default value.

You can specify the value of the **GoalVersionTypeCode** attribute only as **ACTIVE**. if you don't specify any value, the attribute value defaults to **ACTIVE**.

Note: When you load goals using HDL, approvals aren't triggered.

Goal Components

The components that you can supply when loading goals depend on the goal type. This table identifies when you can provide each component.

Goal Component	Goal Type
Goal	All
Goal Action	All
Goal Plan Goal	• Worker performance goals • Organization goals • Template goals
Goal Target Outcome	• Worker goals • Template goals
Goal Measurement	All **Note:** You can specify only one Goal Measurement component for a development goal.
Goal Access	• Worker goals
Goal Alignment	• Worker performance goals • Organization goals

Profile Options

This table identifies profile options that must be both enabled and set to **Y** for successful loading of goal components.

Profile Option	What the Option Enables
HRG_ENABLE_GOAL_ALIGN	Alignment of goals in the Goal Alignment component
HRG_ENABLE_GRANT_ACCESS	Shared goals in the Goal Access component
HRG_ENABLE_OUTCOMES	Creation of goal target outcomes in the Goal Target Outcome component
HRG_ENABLE_TASK	Creation of goal tasks in the Goal Action component

- To enable profile options, use the **Manage Profile Options** task.
- To set the value of a profile option, use the **Manage Administrator Profile Values** task.

Perform both tasks in the Setup and Maintenance work area.

Deleting Goals

You can delete Goal objects using HCM Data Loader. When you delete a parent Goal object, these child components are also deleted automatically:

- Goal Access
- Goal Action
- Goal Alignment, where **GoalId** or **AlignedGoalId** identifies the goal that you're deleting
- Goal Plan Goal
- Goal Target Outcome

You can also delete child components individually, if required.

Goal Measurement components aren't deleted automatically when you delete the parent goal using HCM Data Loader. You must delete Goal Measurement components explicitly before deleting the parent goal. This example Goal.dat file deletes a Goal Measurement component. The parent goal is identified by its source key.

```
METADATA|GoalMeasurement|SourceSystemId|SourceSystemOwner|GoalId(SourceSystemId)
DELETE|GoalMeasurement|GoalMeasurement_001_SSID|VISION|WorkerGoal_001_SSID
```

This example Goal.dat file deletes a goal and its child components, other than Goal Measurement. The goal is identified by its source key.

```
METADATA|Goal|SourceSystemId|SourceSystemOwner
DELETE|Goal|WorkerGoal_001_SSID|VISION
```

Related Topics

- Examples of Loading Goals

Examples of Loading Goals

This topic provides examples that show how to load worker goals, organization goals, and library goals using HCM Data Loader.

Creating a Worker Goal

This example Goal.dat file creates an active performance goal for a worker. In this file:

- The user-key attributes **WorkerPersonNumber** and **WorkerAssignmentNumber** identify the worker and the relevant assignment.
- The user-key attribute **AssignedByPersonNumber** identifies the human resource specialist or line manager who's creating the goal.
- You need to provide a Goal Plan Goal component for performance goals.

Note: For development goals, a Goal Plan Goal component isn't required.

```
METADATA|Goal|SourceSystemId|SourceSystemOwner|GoalName|Description|StartDate|TargetCompletionDate|
WorkerPersonNumber|WorkerAssignmentNumber|StatusCode|GoalSourceCode|AllowKeyAttrUpdateFlag|
IncludeInPerfdocFlag|AssignedByPersonNumber
MERGE|Goal|WorkerGoal_001_SSID|VISION|Increase sales by 20%|Increase sales of Model X123 by conducting more
 promotional campaigns|2020/01/01|2020/12/31|8153787|EEEE8153787|Not started|HR|Y|Y|8153756
METADATA|GoalPlanGoal|SourceSystemId|SourceSystemOwner|GoalId(SourceSystemId)|GoalPlanName|
GoalPlanStartDate|GoalPlanEndDate|PriorityCode
MERGE|GoalPlanGoal|GoalPlanGoal_001_SSID|VISION|WorkerGoal_001_SSID|GP_009_01|2015/01/01|2015/12/31|MEDIUM
```

To specify the users who can delete goals assigned by HR specialists when you upload a performance goal, set the **AllowDelGoalFlag** attribute to one of these values as appropriate:

- ORA_MGR_DEL_GOAL: To allow only managers to delete goals assigned by HR specialists
- ORA_EMP_MGR_DEL_GOAL: To allow both managers and workers to delete goals assigned by HR specialists

This example Goal.dat file creates an active performance goal for a worker where the source is **HR** and managers are allowed to delete the goal.

```
METADATA|Goal|GoalName|StartDate|TargetCompletionDate|WorkerPersonNumber|WorkerAssignmentNumber|
GoalTypeCode|GoalVersionTypeCode|Description|PercentCompleteCode|StatusCode|GoalSourceCode|PrivateFlag|
AllowKeyAttrUpdateFlag|IncludeInPerfdocFlag|RequesterPersonId|AssignedByPersonId|AllowDelGoalFlag
MERGE|Goal|Oct22_HDL_Goal_Test1A_Jan22_00A|2022/01/01|2022/05/01|8153765|EEEE8153765|PERFORMANCE|ACTIVE|
Description new goal|0|Not started|HR|N|Y|Y|100000008153756|100000008153756|ORA_MGR_DEL_GOAL
METADATA|GoalPlanGoal|GoalName|GoalWorkerPersonNumber|GoalWorkerAssignmentNumber|GoalTypeCode|
GoalVersionTypeCode|GoalStartDate|GoalTargetCompletionDate|GoalPlanName|GoalPlanStartDate|GoalPlanEndDate|
PriorityCode
MERGE|GoalPlanGoal|Oct22_HDL_Goal_Test1A_Jan22_00A|8153765|EEEE8153765|PERFORMANCE|ACTIVE|2022/01/01|
2022/05/01|Default Goal Plan - Vision Corporation Enterprise|2022/01/01|2022/12/31|High
```

Creating a Worker Goal Against a Library Goal

This example Goal.dat file creates an active performance goal against a library goal for a worker. HCM Data Loader doesn't take values from the library goal. Instead, you must supply these values when you load the worker goal. The library goal must have already been loaded as a content item. The **ReferenceContentItemName** attribute is the user key for the library goal and identifies the library goal.

```
METADATA|Goal|SourceSystemId|SourceSystemOwner|GoalName|Description|StartDate|TargetCompletionDate|
WorkerPersonNumber|WorkerAssignmentNumber|StatusCode|GoalSourceCode|AllowKeyAttrUpdateFlag|
IncludeInPerfdocFlag|AssignedByPersonNumber|ReferenceContentItemName
MERGE|Goal|WorkerGoal_002_SSID|VISION|Complete Design Thinking training|Design Thinking is a process for
 creative problem solving.|2020/01/01|2020/12/31|8153787|EEEE8153787|Not started|HR|Y|Y|8153756|ZHRG-
Completing Learning Path
```

Creating an Organization Goal

This example Goal.dat file creates and publishes an organization goal for a line manager. Here are some attributes include in this file:

- The user-key attribute **WorkerAssignmentNumber** identifies the line manager who owns the organization goal.
- To publish the organization goal to the organization, set the **PublishedFlag** attribute to **Y**.
- The **GoalPlanGoal** attribute identifies the goal plan to which the goal belongs.

```
METADATA|Goal|SourceSystemId|SourceSystemOwner|PersonId(SourceSystemId)|AssignmentId(SourceSystemId)|
GoalTypeCode|GoalVersionTypeCode|ApprovalStatusCode|GoalName|Description|StatusCode|PriorityCode|
PercentCompleteCode|StartDate|TargetCompletionDate|IncludeInPerfdocFlag|AllowKeyAttrUpdateFlag|
```

```
ActualCompletionDate|SuccessCriteria|LevelCode|CategoryCode|GoalSourceCode|MeasureTypeCode|TargetType|
UomCode|MeasureCalcRuleCode|GoalUrl|PrivateFlag|Comments|WorkerAssignmentNumber|PublishedFlag|
OrganizationName|AssignedByPersonNumber
METADATA|GoalPlanGoal|SourceSystemId|SourceSystemOwner|GoalId(SourceSystemId)|GoalPlanId(SourceSystemId)|
Weighting|PriorityCode|GoalOrganizationName|GoalPlanStartDate|GoalPlanEndDate|GoalName|GoalPlanName|
GoalWorkerAssignmentNumber|GoalWorkerPersonNumber|GoalTypeCode|GoalVersionTypeCode|GoalStartDate|
GoalTargetCompletionDate
MERGE|Goal|Goal_SSID_101_V1|VISION|100000008153756|100000008154060|Performance|ACTIVE||Complete Compliance
 Training|Compliance Training includes modules on ethics, corruption, sexual harrasment.|Not_Started||0|
2020/08/01|2020/12/31|N|Y||||CAREER|WORKER|QUALITATIVE|||||N|||Y||
MERGE|GoalPlanGoal|GoalPlanGoal_SSID_101_V1|VISION|Goal_SSID_101_V1|GP_SSID_101|11|MEDIUM||||||||||
```

Creating a Development Goal

This example Goal.dat file creates an active development goal for a worker. The goal is identified by the source keys.

```
METADATA|Goal|SourceSystemId|SourceSystemOwner|GoalName|Description|StartDate|TargetCompletionDate|
PersonId|AssignmentId|GoalTypeCode|GoalVersionTypeCode|PercentCompleteCode|StatusCode|RequesterPersonId|
AssignedByPersonId|GoalSourceCode
MERGE|Goal|dak_test_002|HCMQA-001|Learn Agile methodologies|Attend the Agile Methodology course and learn
 the 5 stages.|2023/10/01|2023/12/30|100000008153789|100000008154093|DEVELOPMENT|ACTIVE|0|Not started|
100000008153756|100000008153756|Worker
```

Creating a Library Goal

You can create library goals that can be referenced by other goal objects. The measurement details of a library goal are stored separately. To load the goal details and measurement details for a goal, create as many ContentItem lines as you need. Use these values for the `contextName` attribute:

- `GOAL`: To load goal details
- `GOAL_MEASUREMENTS`: To load measurement details

You use ContentItem.dat as in this example to add library goals.

```
METADATA|ContentItem|SourceSystemOwner|SourceSystemId|Name|ContextName|DateFrom|ItemDate1|ItemDate2|
ItemText1|ItemText5|ItemText6|ItemText7|ItemText9|ItemText10|ItemNumber1|ItemNumber2|ItemTextTl14
MERGE|ContentItem|VISION|LGEXGOAL|Library Goal Example Goal|GOAL|2020/04/01|2020/04/01|2020/04/30|A||||
PERFORMANCE|HR_MGR|||<p>Comment for the goal category</p>
MERGE|ContentItem|VISION|LGEXMEAS|Library Goal Example Measurement|GOAL_MEASUREMENTS|2020/04/01|2020/04/01|
2020/04/30||QUANTITATIVE|MAX|CURRENCY|||11|12|
```

To link the goal and its measurement, you need to create a content item relationship. You use ContentItemRelationship.dat as in this example to create a content item relationship.

```
METADATA|ContentItemRelationship|SourceSystemOwner |SourceSystemId|DateFrom|DateTo|Mandatory|
RelationshipCode|ContextName|ContentItemName|RelatedContentTypeContextName|RelatedContentItemName
MERGE|ContentItemRelationship|VISION|SS1_REL1|2020/04/01||N|CHILD|GOAL|Library Goal Example Goal|
GOAL_MEASUREMENTS|Library Goal Example Measurement
```

After you import the .zip file containing the content items, verify if you can add the new library goal.

1. Go to **Me** > **Career and Performance** > **Goals**.
2. In your Goals page, select the review period and goal plan.
3. In the goal plan section, click **Add**.
4. In the Add Goal page, confirm that the recently imported library goals appear in the **Library Goal** list.

Examples of Loading Goal Components

Goals can have tasks and desired outcomes. You can also add measurements to goals. This example shows how you can load these goal components.

This example Goal.dat file creates an active performance goal.

> **Note:** You need to provide a goal plan goal attribute while creating a performance goal for a goal plan. This indicates that the goal is part of a goal plan.

```
METADATA|Goal|SourceSystemId|SourceSystemOwner|WorkerPersonNumber|WorkerAssignmentNumber|GoalTypeCode|
GoalName|Description|StartDate|TargetCompletionDate|PercentCompleteCode|StatusCode|GoalVersionTypeCode|
VersionDate|PriorityCode|AssignedByPersonNumber|GoalSourceCode|PrivateFlag|AccessToHierarchyFlag|
AllowKeyAttrUpdateFlag
MERGE|Goal|Goal_Test_SK2|VISION|8153756|EEEE8153756|PERFORMANCE|Drive customer satisfaction score to 90
 points by end of year|Drive CSP to 90 in three stages|2001/01/01|2099/12/31|0|NOT_STARTED|ACTIVE|2018/11/27
 00:00:00|MEDIUM|8153756|HR|N|N|Y
METADATA|GoalPlanGoal|SourceSystemId|SourceSystemOwner|GoalId(SourceSystemId)|GoalPlanName|
GoalPlanStartDate|GoalPlanEndDate|PriorityCode
MERGE|GoalPlanGoal|GoalPlan_Test_SK2|VISION|Goal_Test_SK2|Default Goal Plan - Vision Corporation Enterprise|
2001/01/01|2099/12/31|HIGH
```

In the example, the `GoalId(SourceSystemId)` attribute identifies the goal and the `GoalPlanName` attribute identifies the goal plan to which this goal is added. Note that the examples for loading the different goal components refer to this goal.

Loading Tasks

Use the **GoalAction** discriminator to load goal task records using HCM Data Loader.

This example Goal.dat file loads a goal task to an active performance goal. The `ActionName` attribute identifies the task, which in this example is **Organize customers conference**. The `GoalId(SourceSystemId)` attribute identifies the goal to which the task is added.

```
METADATA|GoalAction|GoalId(SourceSystemId)|ActionName|ActionTypeCode|StartDate|SourceSystemOwner|
SourceSystemId
MERGE|GoalAction|Goal_Test_SK2|Organize customers conference|CONFERENCE|2014/01/01|VISION|
GoalAction_Test_SK2
```

Loading Measurements

Use the **GoalMeasurement** discriminator to load goal measurement records using HCM Data Loader.

This example Goal.dat file associates a goal measurement with an active performance goal. In this example Goal.dat file, the `MeasurementName` attribute identifies the measurement that's added to the goal, which in this example is **Customer Satisfaction Score**. The `GoalId(SourceSystemId)` attribute identifies the goal to which the measurement is added.

```
METADATA|GoalMeasurement|GoalId(SourceSystemId)|MeasurementName|SourceSystemOwner|SourceSystemId
MERGE|GoalMeasurement|Goal_Test_SK2|Customer Satisfaction Score|VISION|GoalMeasurement_Test_SK2
```

Loading Target Outcomes

Use the **GoalTargetOutcome** discriminator to load target outcome records using HCM Data Loader.

Note: You can load target outcome records for all profile content sections.

This example Goal.dat file creates a target outcome for an active performance goal. In this example Goal.dat file, the `GoalId(SourceSystemId)` attribute identifies the goal to which the target outcome is added. The attribute `SourceSystemId` identifies the target outcome, which in the example is **GoalTarOutProItem_SSID_101_V1** .

```
METADATA|GoalTargetOutcomeProfileItem|ContentItemId(SourceSystemId)|ContentTypeId|CountryId|DateFrom|
DateTo|Importance|InterestLevel|ItemDate1|ItemDate10|ItemDate2|ItemDate3|ItemDate4|ItemDate5|ItemDate6|
ItemDate7|ItemDate8|ItemDate9|ItemDecimal1|ItemDecimal2|ItemDecimal3|ItemDecimal4|ItemDecimal5|
ItemNumber1|ItemNumber10|ItemNumber2|ItemNumber3|ItemNumber4|ItemNumber5|ItemNumber6|ItemNumber7|
ItemNumber8|ItemNumber9|ItemText20001|ItemText20002|ItemText20003|ItemText20004|ItemText20005|ItemText2401|
ItemText24010|ItemText24011|ItemText24012|ItemText24013|ItemText24014|ItemText24015|ItemText2402|
ItemText2403|ItemText2404|ItemText2405|ItemText2406|ItemText2407|ItemText2408|ItemText2409|ItemText301|
ItemText3010|ItemText3011|ItemText3012|ItemText3013|ItemText3014|ItemText3015|ItemText302|ItemText303|
ItemText304|ItemText305|ItemText306|ItemText307|ItemText308|ItemText309|Mandatory|ProfileId(SourceSystemId)|
ProfileItemId|QualifierId1|QualifierId2|RatingLevelId1|RatingLevelId2|RatingLevelId3|RatingModelId1|
RatingModelId2|RatingModelId3|SourceId|SourceKey1|SourceKey2|SourceKey3|SourceType|StateProvinceId|
SourceSystemOwner|SourceSystemId|GUID|StateGeographyCode|StateCountryCode|CountryGeographyCode|
CountryCountryCode|ContentType|ContentItem|ProfileCode|RatingModelCode1|RatingModelCode2|RatingModelCode3|
RatingLevelCode1|RatingLevelCode2|RatingLevelCode3|QualifierCode1|QualifierCode2|QualifierSetCode1|
QualifierSetCode2|ItemClob1File|ItemClob2File|ItemClob3File|ItemClob4File|ItemClob5File|SectionId|
SectionName|GoalId(SourceSystemId)|GoalName|GoalStartDate|GoalTargetCompletionDate|GoalTypeCode|
GoalVersionTypeCode|GoalWorkerPersonNumber|GoalWorkerAssignmentNumber|GoalOrganizationName
MERGE|GoalTargetOutcomeProfileItem|HDLCISSID2HUB19|104||2020/01/01|2020/12/31|2|||||||||||||||||||||||||
GoalTarOutProItem|||||||||||||||||||||||||||||||||||HRTJOBPROFILESSID2|||||||||||||||VISION|
GoalTarOutProItem_SSID_101_V1|||||||||||||||||||||||||Goal_Test_SK2|||||||||
```

Loading Goal Shares

Use the **GoalAccess** discriminator to load goal sharing records using HCM Data Loader. The Goal Access component holds references to the employees the goal is shared with.

This example Goal.dat file creates a goal access record to share the goal with another employee. The `GoalId(SourceSystemId)` attribute identifies the goal that's shared. The `GoalAccessPersonNumber` attribute provides a unique reference to the person that the goal is shared with.

```
METADATA|GoalAccess|GoalId(SourceSystemId)|GoalAccessPersonNumber|GoalActionAccessFlag|SourceSystemOwner|
SourceSystemId
MERGE|GoalAccess|Goal_Test_SK2|8153766|Y|VISION|GoalAction_Test_SK2
```

Loading Goal Alignment

Use the **GoalAlignment** discriminator to load goal alignment records using HCM Data Loader. The Goal Alignment component holds references to the goal with which the goal is aligned.

This example Goal.dat file creates a goal alignment record. The `GoalId(SourceSystemId)` attribute identifies the goal that's aligned. The goal to which it's aligned is identified by the `AlignedGoalId(SourceSystemId)` attribute.

```
METADATA|GoalAlignment|GoalId(SourceSystemId)|AlignedGoalId(SourceSystemId)|SourceSystemOwner|SourceSystemId
MERGE|GoalAlignment|Goal_Test_SK2|Goal_Test_SK1|VISION|GoalAlignment_Test_SK2
```

Guidelines for Loading Goal Plans

Goal plans are used for the mass assignment of goals to worker populations. This topic describes aspects of the Goal Plan object that you must understand to load goal plans successfully.

Assigning Goals to Workers

When you load a Goal Plan object, the plan isn't automatically assigned to workers. You can assign the goal plan to workers in either of these ways:

- Schedule goal plan assignments using the **Manage Goal Plans** task.
- Load the Goal Plan Assignment component of the goal plan.

The Mass Request Component

The Mass Request component identifies who's requesting the goal plan. You can identify the plan requester on the goal plan itself by supplying the **RequestSubmissionDate** and **ReqSubmittedByPersonNumber** attributes, for example. In this case, the Mass Request component is created automatically and you don't have to supply it. However, you must supply a Mass Request component when you create a goal plan:

- If you don't identify the plan requester on the Goal Plan object.
- To associate a source key with the Mass Request record.

  Alternatively, you can update the default source key later using the Source Key object.
- If you're loading child components of the Mass Request component.

  The child components of the Mass Request component identify the employee population to which the goal plan is to be assigned.

Deleting Goal Plans

You can delete Goal Plan objects using HCM Data Loader. However, don't delete goal plans that have been assigned to workers. When you delete a parent Goal Plan object, these child components are also deleted automatically:

- Goal Plan Document Types
- Mass Request

> **Note:** When you delete a Mass Request component, its child components are deleted automatically.

You can also delete child components individually, if required.

Goal Plan Goal and Goal Plan Assignment components aren't deleted automatically when you delete the parent Goal Plan component using HCM Data Loader. You must delete those components explicitly before you delete the parent goal plan. This example GoalPlan.dat file deletes the Goal Plan Goal component that links goal plan GP_001_01 with template goal TemplateGoal_001.

```
METADATA|GoalPlanGoal|SourceSystemId|SourceSystemOwner|GoalPlanExternalId
DELETE|GoalPlanGoal|GPG_001_01_SSID|VISION|GP_001_01
```

This example Goal.dat file deletes the template goal TemplateGoal_001 (SourceSystemId: TemplateGoal_001_SSID).

```
METADATA|Goal|SourceSystemId|SourceSystemOwner
DELETE|Goal|TemplateGoal_001_SSID|VISION
```

This example GoalPlan.dat file deletes the goal plan GP_001_01 and relevant child records.

```
METADATA|GoalPlan|GoalPlanExternalId
DELETE|GoalPlan|GP_001_01
```

Related Topics

- Examples of Loading Goal Plans
- Goal Plans

Examples of Loading Goal Plans

This topic provides examples showing how to load Goal Plan objects using HCM Data Loader.

Creating a Goal Plan

This example GoalPlan.dat file loads a goal plan without a Mass Request component. In this case, a mass request record for the goal plan is created automatically using the values passed for the **ReqSubmittedByPersonNumber** and **RequestSubmissionDate** attributes.

```
METADATA|GoalPlan|GoalPlanExternalId|GoalPlanName|GoalPlanTypeCode|GoalPlanActiveFlag|EnableWeightingFlag|
StartDate|EndDate|EnforceGoalWeightFlag|GoalAccessLevelCode|IncludeInPerfdocFlag|ReqSubmittedByPersonNumber|
RequestSubmissionDate|ReviewPeriodName
MERGE|GoalPlan|GP_001_01|GP_001_01|ORA_HRG_WORKER|A|Y|2015/01/01|2015/12/31|Y|HR specialist and manager|N|
8153756|2015/09/20|Default Review Period - Vision Corporation Enterprise
```

Creating a Goal Plan with Document Types

This example GoalPlan.dat file creates a goal plan with the **IncludeInPerfdocFlag** attribute set to **Y** and assigns a document type. In this example, the **GoalPlanExternalId** attribute in the **GoalPlanDocTypes** component identifies the goal plan the performance document type is for.

```
METADATA|GoalPlan|GoalPlanExternalId|GoalPlanName|GoalPlanTypeCode|GoalPlanActiveFlag|EnableWeightingFlag|
StartDate|EndDate|EnforceGoalWeightFlag|GoalAccessLevelCode|IncludeInPerfdocFlag|ReqSubmittedByPersonNumber|
RequestSubmissionDate|ReviewPeriodName
MERGE|GoalPlan|GP_001_02|GP_001_02|ORA_HRG_WORKER|A|Y|2016/01/01|2016/12/31|Y|HR specialist and manager|Y|
8153756|2016/10/04|Default Review Period - Vision Corporation Enterprise
METADATA|GoalPlanDocTypes|GoalPlanExternalId|DocTypeName
MERGE|GoalPlanDocTypes|GP_001_02|Annual Evaluations
```

Creating a Goal Plan with Mass Request Component

Use the **MassRequest** discriminator to load goal plan mass request records using HCM Data Loader. The Mass Request component holds attributes of the mass request object associated with the goal plan. It's referenced by sub-components that collectively define the employee population or organization to assign the goal plan to.

This example GoalPlan.dat file first loads a goal plan. The **GoalPlanExternalId** attribute in the **MassRequest** component that's later included in the file identifies the goal plan the mass request is for.

```
METADATA|GoalPlan|GoalPlanExternalId|GoalPlanName|GoalPlanTypeCode|GoalPlanActiveFlag|EnableWeightingFlag|
StartDate|EndDate|EnforceGoalWeightFlag|GoalAccessLevelCode|IncludeInPerfdocFlag|ReviewPeriodName
MERGE|GoalPlan|GP_001_03|GP_001_03|Performance|A|Y|2016/01/01|2016/12/31|Y|HR specialist and manager|N|
Default Review Period - Vision Corporation Enterprise
METADATA|MassRequest|GoalPlanExternalId|ReqSubmittedByPersonNumber
MERGE|MassRequest|GP_001_03|8153756
```

Note: The attributes **ReqSubmittedByPersonId** and **RequestSubmissionDate** aren't specified for the **GoalPlan** component in this case.

The Mass Request Assignment component holds information about an organization to associate the goal plan with. Use the **MassRequestAssignment** discriminator to load mass request assignment records for a mass request within a goal plan using HCM Data Loader.

This example GoalPlan.dat file creates a goal plan with a mass request assignment for the mass request.

```
METADATA|GoalPlan|GoalPlanExternalId|GoalPlanName|GoalPlanTypeCode|GoalPlanActiveFlag|EnableWeightingFlag|
StartDate|EndDate|EnforceGoalWeightFlag|GoalAccessLevelCode|IncludeInPerfdocFlag|ReviewPeriodName
MERGE|GoalPlan|GP_001_05|GP_001_05|Performance|A|Y|2016/01/01|2016/12/31|Y|HR specialist and manager|N|
Default Review Period - Vision Corporation Enterprise
METADATA|MassRequest|GoalPlanExternalId|ReqSubmittedByPersonNumber
MERGE|MassRequest|GP_001_05|8153756
METADATA|MassRequestAssignment|OrganizationName|GoalPlanExternalId
MERGE|MassRequestAssignment|Vision Corporation Enterprise|GP_001_05
```

This example GoalPlan.dat file creates a mass request assignment for a goal plan and mass request that already exist. This file uses the user keys attributes to identify the existing goal plan and organization, but you can also use source keys instead.

```
METADATA|MassRequestAssignment|OrganizationName|GoalPlanExternalId
MERGE|MassRequestAssignment|Vision Air|GP_001_05
```

Creating a Goal Plan with Eligibility Criteria

Use the discriminator **EligibilityProfileObject** to load eligibility profile records for a goal plan using HCM Data Loader. The Eligibility Profile Object component includes a reference to an eligibility profile object that defines selection criteria for employees to assign the goal plan to.

This example GoalPlan.dat file creates a goal plan with Mass Request and Eligibility Profile Object components. In the Eligibility Profile Object component, the **GoalPlanExternalId** attribute identifies the goal with which the eligibility profile is associated. In this example, the **Goals_EligibilityProfile_Software Developer 4** eligibility profile is associated with the **GP_001_04** goal plan. This eligibility profile is marked as required.

```
METADATA|GoalPlan|GoalPlanExternalId|GoalPlanName|GoalPlanTypeCode|GoalPlanActiveFlag|EnableWeightingFlag|
StartDate|EndDate|EnforceGoalWeightFlag|GoalAccessLevelCode|IncludeInPerfdocFlag|ReviewPeriodName
MERGE|GoalPlan|GP_001_04|GP_001_04|ORA_HRG_WORKER|A|Y|2016/01/01|2016/12/31|Y|HR specialist and manager|N|
Default Review Period - Vision Corporation Enterprise
METADATA|MassRequest|GoalPlanExternalId|ReqSubmittedByPersonNumber
MERGE|MassRequest|GP_001_04|8153756
METADATA|EligibilityProfileObject|GoalPlanExternalId|EligibilityProfileName|RequiredFlag
MERGE|EligibilityProfileObject|GP_001_04|Goals_EligibilityProfile_Software Developer 4|Y
```

Creating a Goal Plan with Mass Request Hierarchy Component

The Mass Request Hierarchy component holds information about a manager hierarchy or specific employees included in the goal plan. Use the **MassRequestHierarchy** discriminator to load mass request hierarchy records for a goal plan using HCM Data Loader. You can specify any of these values for the **CascadingLevel** attribute:

- **0**: Only to the manager
- **1**: Employees who directly report to the manager
- **999999**: All employees reporting to the manager
- **2**: Manager and employees who directly report to the manager
- **1000000**: Manager and all employees reporting to the manager

This example GoalPlan.dat file loads a goal plan with a hierarchy for the mass request. All employees in the hierarchy of the manager identified by the **ManagerAssignmentNumber** attribute in the Mass Request Hierarchy component are assigned the goal that's identified by the **GoalPlanExternalId attribute**.

```
METADATA|GoalPlan|GoalPlanExternalId|GoalPlanName|GoalPlanTypeCode|GoalPlanActiveFlag|EnableWeightingFlag|
StartDate|EndDate|EnforceGoalWeightFlag|GoalAccessLevelCode|IncludeInPerfdocFlag|ReviewPeriodName
MERGE|GoalPlan|GP_001_06|GP_001_06|Performance|A|Y|2016/01/01|2016/12/31|Y|HR specialist and manager|N|
Default Review Period - Vision Corporation Enterprise
METADATA|MassRequest|GoalPlanExternalId|ReqSubmittedByPersonNumber
MERGE|MassRequest|GP_001_06|8153756
COMMENT Assigning Worker to Goal Plan GP_001_06
METADATA|MassRequestHierarchy|GoalPlanExternalId|ManagerAssignmentNumber|CascadingLevel
MERGE|MassRequestHierarchy|GP_001_06|EEEE8153786|999999
```

This example GoalPlan.dat file creates a mass request hierarchy for a goal plan and mass request that already exist. This example uses the user key attribute to identify the goal plan and manager assignment. You can also use source keys to identify these records.

```
COMMENT Assigning Worker to Goal Plan GP_001_06
METADATA|MassRequestHierarchy|GoalPlanExternalId|ManagerAssignmentNumber|CascadingLevel
MERGE|MassRequestHierarchy|GP_001_06|EEEE8153794|0
```

Creating a Mass Request Exemption Component

You can create a mass request for goal plan assignment but exempt certain employees from the assignment. You do this by using the Mass Request Exemption component, which specifies the employees who are exempted from the goal plan. This component can only be passed for MassRequestHierarchy records where the **CascadingLevel** isn't zero. For example, when the cascading level is Direct reports, All reports, Self and direct reports, or Self and all reports.

Use the **MassRequestExemption** discriminator to load mass request exemption records for a goal plan using HCM Data Loader. This example GoalPlan.dat file creates a goal plan with a mass request exemption for the hierarchy.

```
METADATA|GoalPlan|GoalPlanExternalId|GoalPlanName|GoalPlanTypeCode|GoalPlanActiveFlag|EnableWeightingFlag|
StartDate|EndDate|EnforceGoalWeightFlag|GoalAccessLevelCode|IncludeInPerfdocFlag|ReviewPeriodName
MERGE|GoalPlan|GP_001_07|GP_001_07|Performance|A|Y|2016/01/01|2016/12/31|Y|HR specialist and manager|N|
Default Review Period - Vision Corporation Enterprise
METADATA|MassRequest|GoalPlanExternalId|ReqSubmittedByPersonNumber
MERGE|MassRequest|GP_001_07|8153756
COMMENT Assigning Worker to Goal Plan GP_001_07
METADATA|MassRequestHierarchy|GoalPlanExternalId|ManagerAssignmentNumber|CascadingLevel
MERGE|MassRequestHierarchy|GP_001_07|EEEE8153786|999999
COMMENT Excluding one worker from Goal Plan GP_001_07
METADATA|MassRequestExemption|GoalPlanExternalId|ManagerAssignmentNumber|WorkerAssignmentNumber
MERGE|MassRequestExemption|GP_001_07|EEEE8153786|EEEE8153787
```

This file creates a mass request exemption for an existing goal plan and mass request hierarchy.

```
COMMENT Excluding another worker from Goal Plan GP_012_01
METADATA|MassRequestExemption|SourceSystemId|SourceSystemOwner|WorkerAssignmentNumber|
MassRequestHierarchyId(SourceSystemId)|GoalPlanExternalId
MERGE|MassRequestExemption|MRH_012_02_SSID|VISION|EEEE8153794|MRH_012_01_SSID|GP_012_01
```

The source key for mass request hierarchy `MassRequestHierarchyId(SourceSystemId)` identifies the mass request hierarchy. The **GoalPlanExternalId** attribute identifies the goal plan.

Related Topics

- Guidelines for Loading Goal Plans

Examples of Loading Goal Plan Assignments

This topic provides examples that show how to load goal plan assignment objects using HCM Data Loader. The Goal Plan Assignment component holds information about employee assignments for the goal plan.

Use the **GoalPlanAssignment** discriminator to load goal plan assignment records using HCM Data Loader.

Note: Organization goal plans aren't supported in responsive Goal Management.

Creating a New Goal Plan Assignment Component for an Employee

This example GoalPlan.dat file creates a new goal plan and goal plan assignment for an employee. In this example, the **GoalPlanExternalId** attribute identifies the goal plan that's created and then assigned.

```
METADATA|GoalPlan|GoalPlanExternalId|GoalPlanName|GoalPlanTypeCode|GoalPlanActiveFlag|EnableWeightingFlag|
StartDate|EndDate|EnforceGoalWeightFlag|GoalAccessLevelCode|IncludeInPerfdocFlag|ReqSubmittedByPersonNumber|
RequestSubmissionDate
MERGE|GoalPlan|GP_001_01|GP_001_01|ORA_HRG_WORKER|A|Y|2015/01/01|2015/12/31|N|HR specialist and manager|Y|
8153756|2015/04/01
METADATA|GoalPlanAssignment|SourceSystemId|SourceSystemOwner|GoalPlanExternalId|WorkerAssignmentNumber|
Status
MERGE|GoalPlanAssignment|GPA_001_01|VISION|GP_001_01|EEEE8153787|ACTIVE
```

Creating a Goal Plan Assignment For an Existing Goal Plan

This example GoalPlan.dat file creates a goal plan assignment for a goal plan that already exists. In this example, the **GoalPlanExternalId** attribute identifies the existing goal plan that's assigned.

```
METADATA|GoalPlanAssignment|SourceSystemId|SourceSystemOwner|GoalPlanExternalId|WorkerAssignmentNumber|
Status
MERGE|GoalPlanAssignment|GPA_001_01|VISION|GP_001_01|EEEE8153787|ACTIVE
```

Guidelines for Loading Goal Plan Sets

Goal plan sets are used for the mass assignment of related goal plans to worker populations. This topic describes aspects of the Goal Plan Set object that you must understand to load goal plan sets successfully.

Assigning Goal Plan Sets to Workers

When you load a goal plan set, it's not automatically assigned to workers. You can assign the goal plan set to workers in either of these ways:

- Schedule goal plan set assignments using the **Manage Goal Plan Sets** task.
- Load the Goal Plan Assignment component of the Goal Plan object. You may use this approach, for example, to assign goals following the transfer of workers from one legal entity to another.

The Mass Request Component

The Mass Request component identifies who's requesting the goal plan set. You can identify the plan requester on the goal plan set itself by supplying the **RequestSubmissionDate** and **ReqSubmittedByPersonNumber** attributes, for example. In this case, the Mass Request component is created automatically and you don't have to supply it. However, you must supply a Mass Request component when you create a goal plan set:

- If you don't identify the plan requester on the Goal Plan Set object.
- To associate a source key with the mass request record.

  Alternatively, you can update the default source key later using the Source Key object.
- If you're loading child components of the Mass Request component.

  The child components of the Mass Request component identify the employee population to which the goal plan set is to be assigned.

Deleting Goal Plan Sets

You can delete Goal Plan Set objects using HCM Data Loader. When you delete a parent goal plan set, these child components are also deleted automatically:

- Goal Plan Set Plan
- Mass Request

Note: When you delete a Mass Request component, its child components are deleted automatically.

Don't delete goal plan sets that have already been assigned to workers.

This example GoalPlanSet.dat file deletes a goal plan set and its child components.

```
METADATA|GoalPlanSet|GoalPlanSetExternalId
DELETE|GoalPlanSet|GPS_006_01
```

Related Topics

- Examples of Loading Goal Plan Sets
- Goal Plan Sets

Examples of Loading Goal Plan Sets

This topic provides examples showing how to load Goal Plan Set objects using HCM Data Loader.

Creating a Goal Plan Set with Goal Plans

This example GoalPlanSet.dat file creates a goal plan set and includes two goal plans.

```
METADATA|GoalPlanSet|GoalPlanSetExternalId|GoalPlanSetName|Description|GoalPlanSetActiveFlag|StartDate|
EndDate|ReqSubmittedByPersonNumber|RequestSubmissionDate| ReviewPeriodName
```

```
MERGE|GoalPlanSet|GPS_002_01|GPS_002_01|GPS_002_01|A|2015/01/01|2015/12/31|8153756|2015/09/28|Default Review
 Period - Vision Corporation Enterprise
COMMENT Assign two goal plans without worker or eligibility profile assignment to the goal plan set
METADATA|GoalPlanSetPlan|GoalPlanSetExternalId|GoalPlanExternalId|Weighting
MERGE|GoalPlanSetPlan|GPS_002_01|GP_001_02|60
MERGE|GoalPlanSetPlan|GPS_002_01|GP_001_03|40
```

In this example, the goal plan set has no Mass Request component. Therefore, the mass request record for the goal plan set is created automatically using the **ReqSubmittedByPersonNumber** and **RequestSubmissionDate** attribute values.

Creating a Goal Plan Set with a Mass Request Component

This example GoalPlanSet.dat file creates a goal plan set with a Mass Request component. You don't provide the **ReqSubmittedByPersonNumber** and **RequestSubmissionDate** attributes for the Goal Plan Set component in this case.

```
METADATA|GoalPlanSet|GoalPlanSetExternalId|GoalPlanSetName|Description|GoalPlanSetActiveFlag|StartDate|
EndDate|ReviewPeriodName
MERGE|GoalPlanSet|GPS_001_02|GPS_001_02|GPS_001_02|A|2016/01/01|2016/12/31|Default Review Period - Vision
 Corporation Enterprise
METADATA|MassRequest|GoalPlanSetExternalId|ReqSubmittedByPersonNumber
MERGE|MassRequest|GPS_001_02|8153756
```

Creating a Goal Plan Set with Mass Request and Mass Request Hierarchy Components

This example GoalPlanSet.dat file creates a goal plan set with Mass Request and Mass Request Hierarchy components.

```
METADATA|GoalPlanSet|GoalPlanSetExternalId|GoalPlanSetName|Description|GoalPlanSetActiveFlag|StartDate|
EndDate|ReviewPeriodName
MERGE|GoalPlanSet|GPS_001_04|GPS_001_04|GPS_001_04|A|2016/01/01|2016/12/31|Default Review Period - Vision
 Corporation Enterprise
METADATA|MassRequest|GoalPlanSetExternalId|ReqSubmittedByPersonNumber
MERGE|MassRequest|GPS_001_04|8153756
COMMENT Assigning goal plan set GPS_001_04 to manager and reports
METADATA|MassRequestHierarchy|GoalPlanSetExternalId|ManagerAssignmentNumber|CascadingLevel
MERGE|MassRequestHierarchy|GPS_001_04|EEEE8153786|999999
```

Examples of Loading Performance Documents

You can load performance documents in bulk using HCM Data Loader. This topic shows you how to load Performance Document objects with attachment, participant, section ratings, and comment components. In the examples, all components are referenced by their user keys.

The Operation attribute you specify determines the action to apply. This table lists valid operation values defined in the Performance Administration Action lookup (ORA_HRA_ADMIN_ACTION) and tells you about any dependencies.

Operation Value	Notes
ORA_CREATE_PD	Creates a performance document. This action requires that performance template period and worker assignment exist.
ORA_COMPLETE_PD	Completes a performance document. This action requires that a performance document already exists.

Operation Value	Notes
ORA_CANCEL_PD	Cancels a performance document. This action requires that a performance document already exists.
ORA_DELETE_PD	Deletes a performance document. This action requires that a performance document already exists and its status is Canceled.
ORA_CALC_MANAGER_RATING	Updates manager calculated section ratings in performance documents. This action requires that a performance document already exists.
ORA_ADD_PARTICIPANT	Creates or deletes participants for performance documents. This action requires that a performance document already exists.
ORA_ATTACH_PD	Creates or deletes attachments for performance documents. This action requires that a performance document already exists.
ORA_TRANSFER_PD	Transfers performance documents between managers. This action requires that a performance document already exists and the same performance document doesn't already exist on the assignment that it's being transferred to.
ORA_UPDATE_MODEL_PROFILE_IN_PD	Updates model profile competency sections in performance documents with new competencies. This action requires that a performance document already exists.
ORA_SYNC_PD	Synchronizes goals in performance documents with the employee's goal plan. This action requires that a performance document already exists.
ORA_RATING_PD	Provides section ratings and comments in performance documents. This action requires that a performance document already exists.
ORA_UPDATE_PERS_PROFILE_IN_PD	Synchronizes competencies in person profile competency sections in performance documents with the employee's Talent profile. This action requires that a performance document already exists.
ORA_CALCULATED_RATING_PD	Updates calculated section ratings in performance documents used by All-in-One Evaluations. This action requires that a performance document already exists.
ORA_SUBMIT_PD	Submits manager evaluation tasks in performance documents used by All-in-One Evaluations. This action requires that a performance document already exists.
ORA_SUBMIT_PD_IGNORE_WARNING	Submits manager evaluation tasks and ignores processing warnings in performance documents used by All-in-One Evaluations. This action requires that a performance document already exists.

Loading Performance Documents

This example PerfDocComplete.dat file creates a performance document for a specified worker assignment.

```
METADATA|PerfDocComplete|AssignmentNumber|CustomaryName|StartDate|EndDate|Operation|ManagerAssignmentNumber
MERGE|PerfDocComplete|E121468|Annual Evaluation|2023/01/01|2023/12/31|ORA_CREATE_PD|EEEE8153756
```

The performance template contains default values for start date, end date, and manager assignment number that populate when you create the document. To override the defaults, simply provide your own values in the .dat file.

> **Note:** If you need to mass update participants or attachments, you must delete and load them again using HCM Data Loader.

Loading Attachments

This example PerfDocComplete.dat file loads an attachment to an existing performance document.

```
METADATA|PerfDocComplete|AssignmentNumber|CustomaryName|Operation
```

```
MERGE|PerfDocComplete|EEEE8153772|Annual Evaluation|ORA_ATTACH_PD
METADATA|Attachment|CustomaryName|AssignmentNumber|Title|DataTypeCode|URLorTextorFileName|MimeType|File|
Description
MERGE|Attachment|Annual Evaluation|EEEE8153772|JBlum Resume|FILE|jblum.txt|text/plain|jblum.txt|Jorge Blum
 Resume
```

These rules apply when you load attachments:

- You must include the parent performance document in the PerfDocComplete.dat file. Set the Operation attribute to **ORA_ATTACH_PD** for both MERGE and DELETE instructions for the Attachment component.
- The MimeType value is the Multipurpose Internet Mail Extensions (MIME) value, as defined and standardized by the Internet Engineering Task Force in RFC 6838.
- The attachment itself is a binary large object, referenced by the File attribute. You include the attachment in the BlobFiles folder in the .zip file that you upload.
- You can only bulk delete attachments and load them again. Update action isn't supported.

Loading Participants

This example PerfDocComplete.dat file deletes a participant record from an existing performance document.

```
METADATA|PerfDocComplete|AssignmentNumber|CustomaryName|Operation
MERGE|PerfDocComplete|EEEE8153772|Annual Evaluation|ORA_ADD_PARTICIPANT
METADATA|Participant|AssignmentNumber|CustomaryName|ParticipantRoleTypeCode|ParticipantPersonNumber
DELETE|Participant|EEEE8153772|Annual Evaluation|Colleague|ZHRA-8154961
```

These rules apply when you load participants:

- When you create participants, the parent performance document must already exist. You can't perform two actions on the .dat file. For example, you can’t create performance document and add a participant in the same PerfDocComplete.dat file.
- You must include the parent performance document in the PerfDocComplete.dat file. Set the Operation attribute to ORA_ADD_PARTICIPANT for both MERGE and DELETE instructions for the Participants component.
- The participant person record, as identified by the ParticipantPersonNumber attribute value, must already exist.
- The ParticipantRoleTypeCode value must have been selected in the performance template.
- You can only bulk delete participants and load them again. Update action isn't supported.
- When deleting participants in a performance document, you must set the ForceDelete optional parameter to Y if the feedback status is anything other than Request Not Sent (REQUEST_NOT_SENT) or Not Started (NOTSTR). This additional parameter ensures the user understands that completed feedback is deleted and can't be recovered.
- If you want the participant to be sent a notification to provide feedback as part of the data load, set the Send Notifications to Mass Load Participants process to Yes. You can enter a message for the notification in the Message attribute.

Loading Performance Document Section Ratings and Comments

This example PerfDocComplete.dat file updates existing section ratings and comments of worker, manager, and participants. This applies to all sections where section ratings or comments, or both section ratings and comments, are enabled.

```
METADATA|PerfDocComplete|AssignmentNumber|CustomaryName|Operation
MERGE|PerfDocComplete|EEEE8153805|360 WKR and MGR allow to select and request participants-2017|
ORA_RATING_PD
```

```
METADATA|RatingsAndComments|AssignmentNumber|CustomaryName|ParticipantPersonNumber|ParticipantRoleTypeCode|
SectionName|SectionTypeCode|RatingName|Comments
MERGE|RatingsAndComments|EEEE8153805|360 WKR and MGR allow to select and request participants-2017|8153793|
Manager|ZHRA-Competency Section|REG|ZHRA-Outstanding|Manager Comments
MERGE|RatingsAndComments|EEEE8153805|360 WKR and MGR allow to select and request participants-2017|8153818|
Worker|ZHRA-Competency Section|REG|ZHRA-Expert|Worker Comments
```

These are the general considerations for loading performance document section ratings and comments:

- You can use HDL to load performance document section ratings and comments when more than one performance document needs updating.
- You can only update goal or competency section and overall summary ratings, and section comments.
- Use the update action to enter, change, or remove section ratings and comments.

Related Topics

- HCM Data Loader and Performance Business Objects
- Overview of Performance Documents
- Complete Performance Documents
- Delete Performance Documents
- Overview of Participant Feedback in Performance Documents

Example of Changing the Eligibility for Performance Documents

This example PerformanceDocEligibility.dat file process eligibility for performance documents based on the person assignment and updates the eligibility status table. It prevents the eligibility of excluded assignments.

```
METADATA|PerformanceDocEligibility|PersonNumber|PersonId|AsgNumber|AssignmentId|EffectiveDate
MERGE|PerformanceDocEligibility|12|300000015270011|E12|300000015270026|2023/01/03
```

These rules apply when you process performance document eligibility using HDL:

- It's run for all performance documents included in the PerformanceDocEligibility.dat file.
- It can be used to change the existing eligibility of assignments. For example, it can change eligible to non eligible.

Example of Transferring Performance Documents Between an Employee's Assignments

This example PerfDocUpdate.dat file transfers performance goals and performance documents between an employee's assignments.

```
METADATA|PerfDocUpdate|Operation|ReviewPeriod|NewAssignmentNumber|OldAssignmentNumber
MERGE|PerfDocUpdate|ORA_PD_TRANSFER|FY2023|E8153783-2|EEEE8153783
```

This rule applies when you transfer performance documents:

- When the worker assignment already has the same performance document created on a standard template, then the performance document isn't transferred to the new assignment.

Examples of Loading Check-In Documents

You can upload check-in documents for employees when they're unable to create a check-in and the manager isn't available using HDL.

You can also add a new general discussion topic determined by your organization for managers to discuss career growth or policy changes. If a check-in document is created incorrectly, you can delete the check-in document for the employee using HDL.

General Considerations

These are the general considerations for loading check-in documents:

- A check-in template is created and available.
- The employee is eligible for the check-in document.

Loading Check-In Documents

This example CheckInDocument.dat creates a check-in document for an employee's assignment. The check-in document includes general, performance, and development goal discussion topics.

```
METADATA|CheckInDocument|CheckInTemplateName|ReviewPeriodName|ManagerPersonNumber|WorkerPersonNumber|
WorkerAssignmentNumber|DocumentName|CheckInDate
MERGE|CheckInDocument|REVIEW_TEMPLATE|Default Review Period - Vision Corporation Enterprise|8153799|8153756|
E153756|8153756_2022|2022/01/15 00:00:00
METADATA|DiscussionTopic|CheckInTemplateName|ReviewPeriodName|ManagerPersonNumber|WorkerPersonNumber|
WorkerAssignmentNumber|DocumentName|Name|TopicType|GoalTypeCode|GoalVersionTypeCode|GoalName|StartDate|
TargetCompletionDate|GoalPlanExternalId
MERGE|DiscussionTopic|REVIEW_TEMPLATE|Default Review Period - Vision Corporation Enterprise|8153799|8153756|
E153756|8153756_2022|General Meet|ORA_GENERAL||||||
MERGE|DiscussionTopic|REVIEW_TEMPLATE|Default Review Period - Vision Corporation Enterprise|8153799|
8153756|E153756|8153756_2022||ORA_PERF_GOAL|PERFORMANCE|ACTIVE|Performance Goal Name|2022/01/01|2022/12/31|
100100037213210
MERGE|DiscussionTopic|REVIEW_TEMPLATE|Default Review Period - Vision Corporation Enterprise|8153799|8153756|
E153756|8153756_2022||ORA_DEV_GOAL|DEVELOPMENT|ACTIVE|Development Goal Name|2022/01/01|2022/12/31|
```

Examples of Loading Succession Plans

You can use HCM Data Loader to create and update succession plans. A succession plan identifies a group of workers as candidates to replace key personnel in an organization.

Succession plans include assessments of a worker's readiness to move into a vacancy and the risk and impact of the worker's loss. This topic provides some examples showing how to load succession plans.

Create Succession Plans

This example SuccessionPlan.dat file creates an incumbent succession plan using source keys.

```
METADATA|SuccessionPlan|SourceSystemOwner|SourceSystemId|PlanName|PlanType|Status|AccessTypeCode|
IncumbentPersonId(SourceSystemId)
MERGE|SuccessionPlan|VISION|SP816003|Name of the incumbent succession plan|INCUMBENT|ACTIVE|PRIVATE|
VISION006
METADATA|SuccessionPlanOwner|SourceSystemOwner|SourceSystemId|PlanId(SourceSystemId)|OwnerTypeCode|
PersonId(SourceSystemId)
MERGE|SuccessionPlanOwner|VISION|SPO816003|SP816003|ADMINISTRATOR|VISION013
METADATA|SuccessionPlanCandidate|SourceSystemOwner|SourceSystemId|PlanId(SourceSystemId)|
PersonId(SourceSystemId)|Status|CandidateReadiness
MERGE|SuccessionPlanCandidate|VISION|SPC816003|SP816003|VISION014|ACTIVE|READY_NOW
```

This example SuccessionPlan.dat file creates a job succession plan. It uses user keys to reference the job to which the succession plan relates, the succession plan owner, and its candidates.

```
METADATA|SuccessionPlan|SourceSystemOwner|SourceSystemId|PlanName|PlanType|Status|AccessTypeCode|JobCode|
JobSetCode
MERGE|SuccessionPlan|VISION|SP_MRKT_DIR|Name of the job succession plan|JOB|ACTIVE|PRIVATE|HSDL_MRKT_DIR|
COMMON
METADATA|SuccessionPlanOwner|SourceSystemOwner|SourceSystemId|PlanId(SourceSystemId)|OwnerTypeCode|
PlanOwnerPersonNumber
MERGE|SuccessionPlanOwner|VISION|SPO_MRKT_DIR|SP_MRKT_DIR|ADMINISTRATOR|1023853
METADATA|SuccessionPlanCandidate|SourceSystemOwner|SourceSystemId|PlanId(SourceSystemId)|
PlanCandidatePersonNumber|Status|CandidateReadiness
MERGE|SuccessionPlanCandidate|VISION|SPC009_MRKT_DIR|SP_MRKT_DIR|1023119|ACTIVE|READY_NOW
MERGE|SuccessionPlanCandidate|VISION|SPC008_MRKT_DIR|SP_MRKT_DIR|1032314|ACTIVE|READY_NOW
MERGE|SuccessionPlanCandidate|VISION|SPC007_MRKT_DIR|SP_MRKT_DIR|1023112|ACTIVE|READY_NOW
```

These rules apply when you create a succession plan using HCM Data Loader:

- You must supply a Success Plan Owner component, and the **OwnerTypeCode** attribute must be **ADMINISTRATOR**. A succession plan must always have an administrator owner.
- Incumbents, plan owners, and plan candidates must have active assignments.

Enable Alerts for Succession Plans

A succession plan has an Alerts section. This section has check boxes that control whether notifications are sent or not for assignment changes of either the plan incumbent or candidates. You can upload these alerts related attributes for succession plans:

- **IncumbentRoleChangeFlag**: Indicates whether the job role has changed for the incumbent of a succession plan
- **EnableSuccessorAlert**: Indicates whether an alert notification should be sent when a candidate moves to the role of the succession plan
- **EnableMovedToDiffRoleAlert**: Determines whether an alert notification should be sent when a candidate moves to a role that's different from the role of the succession plan
- **EnableIncumbentJobChangeAlert**: Determines whether an alert notification should be sent when the plan incumbent's role changes

  Note: You can set a value for this attribute only for **Incumbent** type succession plans.

When you upload data about the succession plan owners, use the **EnableAlert** attribute to enable alerts for the plan owner.

When you upload candidate data, you can also upload these attributes:

- **ShowSuccessionStatusFlag**: Determines if banners are shown in the Candidates section when a candidate changes roles.
- **SuccessionStatus**: Indicates the code of the candidate succession status. The values that you can enter are **NULL**, **ORA_HRM_MOVED_TO_DIFF_ROLE**, and **ORA_HRM_SUCCESSOR**. The default value is **NULL**.

This example SuccessionPlan.dat file creates a job succession plan. It enables alerts for these assignment changes:

- Candidate moves to the role of the succession plan
- Candidate moves to a role that's different from the role of the succession plan

It also enables alerts for the plan owner and shows a banner for a candidate to indicate the candidate's role change.

```
METADATA|SuccessionPlan|SourceSystemOwner|SourceSystemId|PlanName|PlanType|Status|AccessTypeCode|JobCode|
JobSetCode|EnableSuccessorAlert|EnableMovedToDiffRoleAlert
MERGE|SuccessionPlan|VISION|SP_MRKT_MGR|Marketing Manager job succession plan|JOB|ACTIVE|PUBLIC|
HDL_MRKT_MGR|COMMON|Y|Y
METADATA|SuccessionPlanOwner|SourceSystemOwner|SourceSystemId|PlanId(SourceSystemId)|OwnerTypeCode|
PlanOwnerPersonNumber|EnableAlert
MERGE|SuccessionPlanOwner|VISION|SPO_MRKT_DIR|SP_MRKT_DIR|ADMINISTRATOR|1023853|Y
METADATA|SuccessionPlanCandidate|SourceSystemOwner|SourceSystemId|PlanId(SourceSystemId)|
PlanCandidatePersonNumber|Status|EmergencySuccessor|CandidateReadiness|ShowSuccessionStatusFlag|
SuccessionStatus
MERGE|SuccessionPlanCandidate|VISION|SPC009_MRKT_DIR|SP_MRKT_MGR|1023119|ACTIVE|Y|READY_NOW|Y|
ORA_HRM_SUCCESSOR
MERGE|SuccessionPlanCandidate|VISION|SPC008_MRKT_DIR|SP_MRKT_DIR|1032314|ACTIVE|READY_IN_2_YEARS|N|NULL
```

Remove Plan Owners

Sometimes workers may no longer be part of the organization. You may want to remove terminated workers who were added as owners from succession plans.

This example SuccessionPlan.dat file is used for removing a terminated plan owner from a succession plan. It uses source keys to identify the succession plan.

```
METADATA|SuccessionPlanOwner|SourceSystemOwner|SourceSystemId|PlanId(SourceSystemId)|PlanOwnerPersonNumber
DELETE|SuccessionPlanOwner|VISION|SPO_MRKT_DIR|SP_MRKT_DIR|1023853
```

Examples of Loading External Candidate Details for Succession Plans

You can include external candidates in succession plans. You first need to create the external candidates before you add them to succession plans. Use the **SuccessionPlanExternalCandidate.dat** file to load external candidate details using HCM Data Loader.

Create External Candidates for Succession Plans

You need to provide these attributes when you upload external candidate details using HCM Data Loader:

- **LastName**: The last name of the external candidate.

- **EmailAddress**: The external candidate's email address. This user key attribute identifies the external candidate.

This example SuccessionPlanExternalCandidate.dat file creates an external candidate record.

```
METADATA|SuccessionPlanExternalCandidate|AddressLine1|AddressLine2|Country|CurrentEmployerName|
CurrentJobTitle|EmailAddress|FirstName|LastName|PhoneNumber|PostalCode|State|TownOrCity
MERGE|SuccessionPlanExternalCandidate|Line 1|Line 2|United States|emp 01|job 01|Colby.morris@xxx.com|Colby|
Morris|9728391|94566|CA|Pleasanton
```

This example SuccessionPlanExternalCandidate.dat file creates an external candidate record using source keys.

```
METADATA|SuccessionPlanExternalCandidate|AddressLine1|AddressLine2|Country|CurrentEmployerName|
CurrentJobTitle|EmailAddress|FirstName|LastName|PostalCode|State|TownOrCity|SourceSystemId|SourceSystemOwner
MERGE|SuccessionPlanExternalCandidate|Line 1|Line 2|United States|First Software|Senior Manager|
James_Ng@xxx.com|James|Ng|30214|GA|Fayette|JAMES_NG|VISION
```

Add External Candidates to a Succession Plan

You can use the Succession Plan Candidate component to add external candidates to a succession plan. This table describes the attributes for external candidates.

Attribute Name	Label	Description	Type
ExternalCandidateEmailAddress	External Candidate Email Address	The email address of the external candidate	String
ExternalCandidateId	External Candidate ID	A unique reference to the external candidate	Long

This example SuccessionPlan.dat file shows how you can add external candidates to a succession plan. When you don't use source keys, the external candidate's email ID is the key to fetch the external candidate details.

```
METADATA|SuccessionPlan|SourceSystemOwner|SourceSystemId|PlanName|PlanType|Status|AccessTypeCode|JobCode|
JobSetCode
MERGE|SuccessionPlan|VISION|SP_MRKT_DIR|Plan 01|JOB|ACTIVE|PRIVATE|HDL_MRKT_DIR|COMMON

METADATA|SuccessionPlanCandidate|SourceSystemOwner|SourceSystemId|PlanId(SourceSystemId)|
ExternalCandidateId(SourceSystemId)|ExternalCandidateEmailAddress|Status|EmergencySuccessor|
CandidateRanking|PlanName|PlanCandidatePersonNumber|CandidateReadiness|ShowSuccessionStatusFlag|
SuccessionStatus
MERGE|SuccessionPlanCandidate||||Colby.morris@xxx.com|Active|Yes|1|Plan 01||||
MERGE|SuccessionPlanCandidate|VISION|MRKT_DIR_JAMES_NG|SP_MRKT_DIR|JAMES_NG||Active|Yes|2|||||
```

Delete External Candidate Details

You can delete the details of an external candidate only if the candidate isn't added to any succession plan.

In this example SuccessionPlanExternalCandidate.dat file, the details of the external candidate **Bala Gupta** are deleted.

```
METADATA|SuccessionPlanExternalCandidate|SourceSystemOwner|SourceSystemId
DELETE|SuccessionPlanExternalCandidate|VISION|BALA_GUPTA
```

Examples of Loading Talent Pools

You can use HCM Data Loader to create and update talent pools. You use talent pools to manage the development of the pool members for specific purposes.

Create a Talent Pool

A talent pool can reference any combination of job, job family, model profile, position, department, and grade. These objects, which can define the purpose of the talent pool, have no effect on its processing. However, they must exist in the target environment before you load a talent pool that references them. This example TalentPool.dat file creates a talent pool for a job. Source keys are used to identify both the talent pool and the associated job.

```
METADATA|TalentPool|SourceSystemOwner|SourceSystemId|PoolName|PoolTypeCode|Description|
OwnerPersonId(SourceSystemId)|Status|JobId(SourceSystemId)
MERGE|TalentPool|VISION|TP_T1|Job Chief Exec|TALENT|Talent pool description|1008|A|93079
```

Always set the **PoolTypeCode** attribute to **TALENT**. You must include the **Status** attribute. If you supply no value, then the default status is **A** (Active). Set the status to **I** (Inactive) to inactivate the talent pool.

The Talent Pool component defines the talent pool and the initial pool owner. To identify additional pool owners, you use the Talent Pool Owner component. Additional pool owners are optional, but multiple owners are recommended. Talent pool owners can see talent pools in the application. Use the Talent Pool Member component to define the pool members. This example TalentPool.dat file creates a talent pool with pool members. Source keys are used to identify the talent pool and reference foreign objects.

```
METADATA|TalentPool|SourceSystemOwner|SourceSystemId|PoolName|PoolTypeCode|Description|
OwnerPersonId(SourceSystemId)|Status
MERGE|TalentPool|VISION|TP_T2|Member Talent Pool|TALENT|Talent pool description|1011|A
METADATA|PoolMember|SourceSystemOwner|SourceSystemId|PoolId(SourceSystemId)|MemberId(SourceSystemId)
MERGE|PoolMember|VISION|TP_PM_T2_1012|TP_T2|1012
MERGE|PoolMember|VISION|TP_PM_T2_1013|TP_T2|1013
```

Translate a Talent Pool Name

This example TalentPoolTranslation.dat file translates the name of an existing talent pool. It identifies the existing record by its source key.

```
METADATA|TalentPoolTranslation|SourceSystemOwner|SourceSystemId|PoolName|Language
MERGE|TalentPoolTranslation|VISION|TP_T1|Spanish Translated Pool Name|ES
MERGE|TalentPoolTranslation|VISION|TP_T1|French Translated Pool Name|FR
```

This example TalentPoolTranslation.dat file translates the name of an existing talent pool. It identifies the existing record by its user key.

```
METADATA|TalentPoolTranslation|BasePoolName|PoolStatus|PoolName|Language
MERGE|TalentPoolTranslation|Job Chief Exec|A|Spanish Translated Pool Name|ES
MERGE|TalentPoolTranslation|Job Chief Exec|A|French Translated Pool Name|FR
```

Delete Talent Pools

You can delete a talent pool using HCM Data Loader. You can also use HCM Data Loader to manage talent pools in these ways:

- Make the talent pool inactive.

- Delete talent pool owners, provided that at least one pool owner remains.
- Delete Talent Pool Member components.

This example TalentPool.dat file deletes an existing Talent Pool Owner component.

```
METADATA|PoolOwner|SourceSystemOwner|SourceSystemId|PoolName|PoolOwnerId
DELETE|PoolOwner|VISION|TP_PO_T2_3423|Top Talent|521528
```

This example deletes the Talent Pool, itself. When you delete this parent object, you also delete its children, such as Talent Pool Owner.

```
METADATA|TalentPool|PoolName|PoolTypeCode|Status
DELETE|TalentPool|High Potential|TALENT|A
```

Related Topics

- Talent Pools

Overview of Loading Talent Pool Security Profiles

You can use security profiles to control access to nonprivate talent pools.

Use the **TalentPoolSecurityProfile.dat** file to upload data for the **Talent Pool Security Profile** business object. You can use the **SourceSystemId** and **SourceSystemOwner** attributes to uniquely identify the record.

The **Talent Pool Security Profile** business object has these child components:

- **Talent Pool Security Profile Business Unit**: Uses the **PoolSecProfBusinessUnit** discriminator to identify the talent pool security profile configuration by business units. You can upload security profile configuration for business units only if the value of the **SecureByBusinessUnit** attribute of the parent **Talent Pool Security Profile** business object is **Yes**.
- **Talent Pool Security Profile Department**: Uses the **PoolSecProfDepartment** discriminator to identify the talent pool security profile configuration by departments. You can upload security profile configuration for departments only if the value of the **SecureByDepartment** attribute of the parent **Talent Pool Security Profile** business object is **Yes**.
- **Talent Pool Security Profile Job Family**: Uses the **PoolSecProfJobFamily** discriminator to identify the talent pool security profile configuration by job families. You can upload security profile configuration for job families only if the value of the **SecureByJobFamily** attribute of the parent **Talent Pool Security Profile** business object is **Yes**.

Related Topics

- Talent Pool Security Profiles

Examples of Loading Talent Pool Security Profiles

You can create and update talent pool security profiles using HCM Data Loader (HDL). You can also delete child objects of a talent pool security profile.

Use the **TalentPoolSecurityProfile.dat** file to upload data for the **Talent Pool Security Profile** business object.

Create a Talent Pool Security Profile

In this example, the **HDL-POOL-12** security profile is created that secures talent pools by business unit, department, and job family.

```
METADATA|TalentPoolSecurityProfile|EnabledFlag|Name|SecureByBusinessUnit|SecureByDepartment|
SecureByJobFamily
METADATA|PoolSecProfBusinessUnit|Name|BusinessUnitName
METADATA|PoolSecProfDepartment|Name|DepartmentName
METADATA|PoolSecProfJobFamily|Name|JobFamilyCode

MERGE|TalentPoolSecurityProfile|Y|HDL-POOL-12|Y|Y|Y
MERGE|PoolSecProfDepartment|HDL-POOL-12|ZBENELIG_US_DEPT001
MERGE|PoolSecProfDepartment|HDL-POOL-12|ZBENELIG_US_DEPT002
MERGE|PoolSecProfBusinessUnit|HDL-POOL-12|ZBENELIG_US_Marketing_BU
MERGE|PoolSecProfBusinessUnit|HDL-POOL-12|ZBENELIG_US_Sales_BU
MERGE|PoolSecProfJobFamily|HDL-POOL-12|ZFRCE_PRODUCTION
MERGE|PoolSecProfJobFamily|HDL-POOL-12|ZFRCE_MARKETING
```

Update a Talent Pool Security Profile

In this example, the **HDL-POOL-12** security profile is updated so that it doesn't secure talent pools by job family.

```
METADATA|TalentPoolSecurityProfile|EnabledFlag|Name|SecureByBusinessUnit|SecureByDepartment|
SecureByJobFamily
MERGE|TalentPoolSecurityProfile|Y|HDL-POOL-12|Y|Y|N
```

Delete Child Objects of a Talent Pool Security Profile

In this example, the **ZBENELIG_US_DEPT001** department, the **ZBENELIG_US_Marketing_BU** business unit, and the **ZFRCE_MARKETING** job family code are removed from the **HDL-POOL-12** security profile.

```
METADATA|PoolSecProfBusinessUnit|Name|BusinessUnitName
DELETE|PoolSecProfDepartment|HDL-POOL-12|ZBENELIG_US_DEPT001

METADATA|PoolSecProfDepartment|Name|DepartmentName
DELETE|PoolSecProfBusinessUnit|HDL-POOL-12|ZBENELIG_US_Marketing_BU

METADATA|PoolSecProfJobFamily|Name|JobFamilyCode
DELETE|PoolSecProfJobFamily|HDL-POOL-12|ZFRCE_MARKETING
```

Related Topics

- Talent Pool Security Profiles
- Overview of Security Profiles for Talent Pools

Overview of Enhanced Talent and Model Profiles

Unlike most objects that support bulk-loading by HCM Data Loader, talent and model profiles are highly configurable. How you load data, what sections are applicable and how attributes are validated depend on the configuration of person or model profile.

Use the **Profiles** work area to review the configuration of the person talent and model profiles you're bulk loading data for. Use the **View Business Objects** task in the **Data Exchange** work area to review and map the attributes to section content properties.

To upload talent and model profiles data download:

- Enhanced Talent Profile business object
- Translation objects: Enhanced Talent Profile Translation, Talent Model Profile Extra Info Translation, and Talent Bookmark Translation to create or update translations for certain profile attributes.

Sections and Ratings

The Enhanced Talent Profile object hierarchy is complex. It provides a separate component for each content section and talent rating applicable to person talent and model profiles, such as Areas of Study, Languages, and Talent Score.

Some components are only applicable to person talent profiles while others are only applicable to model profiles. Review the guidelines for each profile type. However, the shape of the profile depends on the configuration of the profile.

How to Identify Applicable Sections

1. Navigate to **Profiles** > **Profile Types**.
2. Click on the type of profile, such as person, job, position, or organization.
3. The Content Sections table lists the sections configured for the type of profile. Only sections that are active here can be bulk-loaded.

   > **Note:** The template name of each section matches the name of the Enhanced Talent Profile component used to load that section. For example, the Degrees section uses the Education template, so use the Education component to bulk load data for the Degrees section.

4. For person profiles, the Profile Talent Ratings table lists ratings that are available for the person talent profile. Not all of these can be loaded using HCM Data Loader.
5. For job, position and organization model profiles, additional information such as description, responsibilities, qualifications, and criticality can be loaded.

   > **Note:** The description, responsibilities, and qualifications can be loaded using the Model Profile Extra Info component in the Enhanced Talent Profile object hierarchy. Criticality information is loaded using the Model Profile Criticality component.

Section Context

A section template can be used to create multiple sections, so when you load data you must define which section your data is for.

The SectionContext attribute is found on all section-related components in the Enhanced Talent Profile object hierarchy. To find the value used to uniquely identify each section:

1. Navigate to **Profiles** > **Profile Types**.
2. Click on the type of profile, such as person, job, position, or organization.
3. Click on the name of the section.
4. Scroll down to below the Content Section Properties table.
5. The value of the DFF Context Code should be supplied to the SectionContext attribute.

Section Properties

Each component in the Enhanced Talent Profile object hierarchy provides all attributes supported by a section. However, your person or model profile might only enable a subset of these. To identify the supported properties:

1. Navigate to **Profiles** > **Profile Types**
2. Click on the type of profile, such as person, job, position, or organization.
3. Click on the name of the section.
4. Review the properties in the Content Section Properties table.

   The property label in this table will usually match the HDL attribute label. However, when content items validation is supported, multiple attributes are often available to provide the property value, either as a free text value or to reference the content item.
5.

Field Type	Validation
Item Value Set	Validated with a content type value set, which is named in the Value Set Name column.
Text	Free-text
Decimal	Numeric value
Date	Date, which needs to be supplied in a specific format
Lookup	Validated with a lookup, which is named in the Value Set Name column.
List of Values	Validated by a different type of list of values. This might be a rating model, usually identified by the value set name, such as Language Rating Model.

   How to find valid values for the different field types is explained below.

   If you supply attribute values for properties not applicable to your section, they might still be loaded. The values aren't visible on the user interface.

Content Items

Preconfigured content sections often use content items to validate the content properties, for example, the language, education establishment, or honor. When you create your own sections, you can choose to use a content item, or a free-text field. Many components in the Enhanced Talent Profile object hierarchy offer both options; a free-text attribute (CompetencyName, LanguageName, and so on) and attributes to capture the content item (ContentItemId or ContentItemName).

Some Enhanced Talent Profile components have multiple references to content items so attribute names might vary. For example, the Education component supplies attributes to reference the education content item but also the educational establishment. Always refer to the attributes labels and descriptions using the View Business Objects task for clarification.

To review valid content items in Item Catalogs:

1. Navigate to **Profiles** > **Item Catalogs**.

2. Select the section template name and specify the item catalog name.
3. Click **Search**.
4. Click on the item catalog name to review the list of content item names that can be supplied to the ContentItemName attribute.

Rating Models and Levels

Many sections enable you to record ratings. Ratings have two parts; the rating model and the rating level. If you're supplying a rating level, you must also specify which rating model the level is using.

To find the rating model used to validate a rating level:

1. Navigate to **Profiles** > **Profile Types**.
2. Click on the type of profile, such as person, job, position, or organization.
3. Click on the name of the section.
4. Review the properties in the Content Section Properties table. The value set name generally has a suffix 'Rating Model' for properties validated by rating models and levels. You might notice some difference here for custom rating models that don't follow the naming standard.

The value displayed in the Value Set Name column isn't the value you supply when bulk-loading data. To find the rating model code:

1. Navigate to **Profiles** > **Profile Rating Models**.
2. Search for the name seen in the Value Set Name column for the Content Section property.
3. Supply the code, such as PERFORMANCE, RISK, or LANGUAGE to the rating model code attribute.
   Tip: The RISK rating model is used for model profile criticality.

To find the valid rating levels:

1. Click on the rating model and click **Edit**.
2. Scroll down to the Rating Levels tab.
3. Supply the rating level value.
   Tip: Use the View Business Objects task to find the name of the attributes used to supply the rating model and rating level. Review the user key of attributes that refer to the RatingLevel integration object name.

Lookups

When a content section property has a field type of Lookup, the Value Set Name column specifies the lookup type. To find the valid lookup codes for a lookup type:

1. Navigate to Setup and Maintenance.
2. Search for and click on the **Manage Common Lookups** task.
3. Search for the lookup using the lookup type.
4. Select the correct lookup type from the Search Results table.
5. Supply any of the lookup codes to the lookup validated attribute.

Guidelines for Loading a Person Talent Profile

The Enhanced Talent Profile object hierarchy only supports the Profiles flow. You can't bulk load data contributed by other flows, such as Learning, Recruiting, Goals, Career Development, Performance Management, or Talent Review.

Only the following components of the Enhanced Talent Profile object hierarchy are applicable to person talent profiles.

- Enhanced Talent Profile
  - Career Statement
  - Career Preference
  - Work Preference
  - Areas of Study
  - Education Level
  - Highest Education Level
  - Technical Post Detail
  - Previous Employment
  - Accomplishment
  - Skill
  - Language
  - Membership
  - Special Project
  - Competency
    - Behavior
  - Certification
    - Certification Attachments
  - Education
    - Education Attachments
  - Honor
    - Honor Attachments
  - Advancement Readiness
  - Risk of Loss
  - Impact of Loss
  - Potential Rating
  - Talent Score
  - Custom Talent Rating
  - Performance Rating
  - Profile Keyword
  - Profile Tag
  - Bookmark
  - Profile Attachments

The table below lists the leading attributes that need to be provided depending on whether the content section is configured as freeform or non-freeform on the Profile Types setup page.

Section	Non-Freeform Leading Attribute	Freeform Leading Attribute

Accomplishments	Not supported	Accomplishment
Areas of Study	Not supported	AreaOfStudy
Licenses and Certifications	ContentItemId/ContentItemName	CertificationName
Competencies	ContentItemId/ContentItemName	CompetencyName
Degrees	ContentItemId/ContentItemName	DegreeName
Education Levels	ContentItemId/ContentItemName	Not supported
Honors and Awards	ContentItemId/ContentItemName	Name
Languages	ContentItemId/ContentItemName	LanguageName
Memberships	ContentItemId/ContentItemName	AffiliationName
Career Preferences	No leading attribute	No leading attribute
Career Statement	No leading attribute	No leading attribute
Work Requirements	No leading attribute	No leading attribute
Previous Employment	Not supported	EmployerName
Technical Post Details	ContentItemId/ContentItemName	Not supported
Highest Education Level	Not supported	HighestEduLevel
Behaviors	ContentItemId/ContentItemName	Not supported
Skills	Not supported	Skills
Special Projects	ContentItemId/ContentItemName	ProjectName

Identifying an Existing Person Talent Profile

To find the profile code of a talent profile, refer to the Cloud Customer Connect HCM Integrations Resource Sharing Center topic *BI Publisher Report: Talent Profile Codes*. This report extracts the profile code that uniquely identifies the person talent profile.

Example of Loading Legacy Talent Profiles

If you've uploaded your employees during data-migration, you might have disabled the automatic generation of talent profiles.

You can use the EnhancedTalentProfile.dat file to load legacy person talent profiles data. You are recommended to use source keys when creating new records, to identify the profile record uniquely.

This example creates a person talent profile record with an accomplishment and certification record for the profile.

```
METADATA|EnhancedTalentProfile|SourceSystemOwner|SourceSystemId|ProfileCode|PersonNumber|ProfileTypeCode|
ProfileStatusCode|ProfileUsageCode|Description|Summary
MERGE|EnhancedTalentProfile|VISION|LD_PRG_PP|Pers_4321|4321|PERSON|A|P|Talent Profile for person number
 4321|Talent Profile for person number 4321
```

```
METADATA|TalentAccomplishment|SourceSystemOwner|SourceSystemId|ProfileId(SourceSystemId)|SectionContext|
Accomplishment|DateFrom|IsLinkedIn|AccomplishmentType
MERGE|TalentAccomplishment|VISION|LD_PRG_PP_ACC|LD_PRG_PP|PERSON_ACCOMPLISHMENT|Reduced operating costs by
 15 percent|2022/01/01|N|COST_REDUCE

METADATA|TalentCertification|SourceSystemOwner|SourceSystemId|ProfileId(SourceSystemId)|SectionContext|
ContentItemId(SourceSystemOwner)|ContentItemId(SourceSystemId)|DateFrom|IsLinkedIn|CertificationNumber
MERGE|TalentCertification|VISION|LD_PRG_PP_CERT|LD_PRG_PP|PERSON_CERTIFICATION|FUSION|300100007702089|
2022/02/01|N|ORA_4321
```

Example of Creating Profile Items for an Existing Talent Profile

For auto-generated talent profiles, you're unlikely to know the source keys used to uniquely identify each record. You can extract these using the Integration Object User Key Map extract, but the example provided here uses user keys.

Use the EnhancedTalentProfile.dat file to load person talent profile data.

This example creates an education and an honor record for an existing person Talent Profile.

```
METADATA|EnhancedTalentProfile|ProfileCode
MERGE|EnhancedTalentProfile|PERS_300100585049120

METADATA|TalentEducation|ProfileCode|SectionContext|DateFrom|ContentItemName
MERGE|TalentEducation|PERS_300100585049120|PERSON_DEGREE|2019/10/02|Juris Doctor

METADATA|TalentHonor|ProfileCode|SectionContext|DateFrom|ContentItemName
MERGE|TalentHonor|PERS_300100585049120|PERSON_HONOR|2019/10/02|Dale Carnegie
```

Guidelines for Loading a Model Profile

Model profiles are never auto-generated. Use HCM Data Loader to create, maintain, relate, and delete model profiles for jobs, positions, and organizations.

Only the following components of the Enhanced Talent Profile object hierarchy are applicable to Model Profiles.

- Enhanced Talent Profile
  - - Model Profile Extra Info
    - Model Profile Criticality
    - Work Preference
    - Accomplishment
    - Skill
    - Language
    - Membership
    - Special Project
    - Competency
      - Behavior
    - Certification
    - Education

- Honor
- Profile Attachments

Model Profile Relationship

Example of Creating a Model Profile

Use the Enhanced Talent Profile object hierarchy to create model profiles.

This example creates a job model profile using the following records:

File Discriminator	Description
EnhancedTalentProfile	Provides a code, name, and description for the model profile.
TalentModelProfileExtraInfo	Creates the job description, responsibility, and qualifications.
TalentCriticality	Defines the criticality of the model profile.
TalentLanguage	Creates the English language requirements, referencing the language content item by source key.
TalentCompetency	Creates the ADA development competency requirements.
TalentProfileRelation	Relates to model profile with the Lead Programmer job.

Use the EnhancedTalentProfile.dat file to supply this data. The attachment files for the job description, responsibility, and qualifications must be supplied in subfolder of the zip file, named ClobFiles.

```
METADATA|EnhancedTalentProfile|SourceSystemOwner|SourceSystemId|ProfileCode|ProfileTypeCode|
ProfileStatusCode|ProfileUsageCode|Description|Summary
MERGE|EnhancedTalentProfile|VISION|LD_PRG_MP|LD_PRG|JOB|A|M|Lead programmer|Profile for lead programmer job.

METADATA|TalentModelProfileExtraInfo|SourceSystemOwner|SourceSystemId|ProfileId(SourceSystemId)|Description|
Responsibilities|Qualifications
MERGE|TalentModelProfileExtraInfo|VISION|LD_PRG_MP_EI|LD_PRG_MP|LeadProgrammerDescription.txt|
LeadProgrammerResponsibilities.txt|LeadProgrammerQualifications.txt

METADATA|TalentCriticality|SourceSystemOwner|SourceSystemId|ProfileId(SourceSystemId)|DateFrom|
CriticalityReason1|CriticalityReason2|CriticalityReason3|CriticalityReason4|CriticalityReason5|
RatingModelCode|JobOrPositionRiskCode|JobOrPositionRiskComments|RequiresSuccessionPlan
MERGE|TalentCriticality|VISION|LD_PRG_MP_CRIT|LD_PRG_MP|2022/01/01|ORGSTRUCTR|MKTVOLATIL|PLANREQ|SCARCE||
RISK|MED|If there is a vacancy for this job, it can increase the risk that your business can miss upcoming
 deadlines.|N

METADATA|TalentLanguage|SourceSystemOwner|SourceSystemId|ProfileId(SourceSystemId)|DateFrom|SectionContext|
IsLinkedIn|ContentItemId(SourceSystemId)|ContentItemId(SourceSystemOwner)|LanguageName|ReadingModelCode|
ReadingLevelCode|SpeakingModelCode|SpeakingLevelCode|WritingModelCode|WritingLevelCode|AbleToTranslateFlag|
AbleToTeachFlag|RequiredFlag|Importance
MERGE|TalentLanguage|VISION|LD_PRG_MP_LAN-ENG|LD_PRG_MP|2022/01/01|JOB_LANGUAGE|N|109000016|VISION||
LANGUAGE|HIGH|LANGUAGE|HIGH|LANGUAGE|HIGH|N|N|Y|

METADATA|TalentCompetency|SourceSystemOwner|SourceSystemId|ProfileId(SourceSystemId)|DateFrom|
SectionContext|ContentItemName|CompetencyName|MinProfModelCode|MinProfLevelCode|MaxProfModelCode|
MaxProfLevelCode|Weight|MinWeight|RequiredFlag|Importance
MERGE|TalentCompetency|VISION|LD_PRG_MP_COM-ADA|LD_PRG_MP|2022/01/01|JOB_COMPETENCY|ADA development||
PROFICIENCY|3|PROFICIENCY|5|80|50|N|
```

```
METADATA|TalentProfileRelation|SourceSystemOwner|SourceSystemId|ProfileId(SourceSystemId)|
ObjectEffectiveStartDate|ObjectEffectiveEndDate|RelationCode|JobCode|JobSetCode
MERGE|TalentProfileRelation|VISION|LD_PRG_MP_R1|LD_PRG_MP|2022/01/01||JOB|LD_PRG|COMMON
```

Guidelines for Loading Core Skills

The Skills Assignment object hierarchy helps to assign core skills to workers in bulk. You assign core skills to people within an organization identified by the manager of that organization.

The object hierarchy consists of the following objects:

- Skills Assignment

  This business object contains the high-level details for a core skill assignment. This includes details like unique name for the skill assignment, the manager of the organization to assign skills to and an action to specify if the intention is to assign or unassign a skill.

  - Core Skill

    Specify the skill requirements for the assignees defined by the Assignee object. Core Skills capture details like name of the skill getting assigned or unassigned and the proficiency level needed for the skill.
  - Assignee

    Core skills are assigned based on the target worker population identified using by the manager of the organization and the assignee type.

    All the assignee records within one SkillsAssignment need to have the same Assignee type. The following assignee types are supported:

    - Job: Specifies any jobs in the application, regardless of whether they have job profiles associated, so the skills will be assigned to anyone within the manager's organization who has this job. When using the Job assignee type, supply the JobId attribute with the surrogate ID or an integration key value that identifies the job, or supply the user key attributes for JobId, viz JobCode and SetCode.
    - Person: Specifies people reporting to to the manager through a primary assignment. For this assignee type, mention the PersonId or user keys of PersonId like PersonNumber.
    - Direct Reports: Only the direct reports of the specified manager.
    - Organization: All of the manager's organization or that of any of the lower-level managers reporting to the manager.

    For assigning skills by a job within a specific team, use the Job assignee type, and populate PersonId of lower-level manager. This will assign the specified core skills to all the workers in lower-level manager's organization with matching job.

Use the Team Skills Center work area and Assign Core Skills page to review the data you want to bulk load for.

Post Processing

After the core skills are loaded, the scheduled process Propagate Dynamic Skills to Workers gets submitted by default.

If this default behavior needs to be changed, navigate to Configure HCM Data Loader task and review the Business Object Post Processes page. Search for the Skills Assignment business object and override the value to No. This will

prevent the scheduled job from getting auto submitted after each skills assignment data load. After the override you need to manually run the scheduled job once Skills Assignment data loads are complete.

Related Topics

- Assigning Core Skills

Example of Assigning Skills to a Manager's Direct Reports

The example assigns skills to person assignees.

```
METADATA|SkillsAssignment|SkillsAssignmentName|Action|AssignedByPersonNumber
MERGE|SkillsAssignment|Directs-Java-SQL|ORA_ASSIGN|3001000

METADATA|Assignee|AssigneeType|SkillsAssignmentName|PersonNumber
MERGE|Assignee|ORA_CORE_DIRECTS|Directs-Java-SQL|3001005
MERGE|Assignee|ORA_CORE_DIRECTS|Directs-Java-SQL|3001006

METADATA|CoreSkill|Skill|SkillsAssignmentName|RequiredProficiencyLevelCode
MERGE|CoreSkill|JAVA|Directs-Java-SQL|3
MERGE|CoreSkill|SQL|Directs-Java-SQL|4
```

Example of Assigning Skills to all People in a Specific Job

The example assigns skills to job assignees.

```
METADATA|SkillsAssignment|SkillsAssignmentName|Action|AssignedByPersonNumber
MERGE|SkillsAssignment|AssignProgrammer_Python_Mongo|ORA_ASSIGN|10469545

METADATA|Assignee|AssigneeType|SkillsAssignmentName|JobCode|SetCode
MERGE|Assignee|ORA_CORE_JOB|AssignProgrammer_Python_Mongo|APPS_DEV_1|US_SET
MERGE|Assignee|ORA_CORE_JOB|AssignProgrammer_Python_Mongo|APPS_DEV_2|US_SET

METADATA|CoreSkill|Skill|SkillsAssignmentName|RequiredProficiencyLevelCode
MERGE|CoreSkill|Python|AssignProgrammer_Python_Mongo|3
MERGE|CoreSkill|MongoDB|AssignProgrammer_Python_Mongo|4
```

This assigns the Core skill Python and MongoDB to all workers in the organization of person 10469545, with job APPS_DEV_1 and APPS_DEV_2.

Example of Assigning Skills to People in a Specific Job Within a Specified Organization

This example assigns the Python and MongoDB skills to all people in the APP_DEV_1 and APPS_DEV_2 jobs who report to the manager identified by person number 3001005. This use case arises when a manager at a higher level in the manager hierarchy wants to assign skills to a set of workers, part of a lower level manager in the hierarchy, identified by their jobs.

```
METADATA|SkillsAssignment|SkillsAssignmentName|Action|AssignedByPersonNumber
MERGE|SkillsAssignment|AssignProgrammer_Python_Mongo|ORA_ASSIGN|10469545

METADATA|Assignee|AssigneeType|SkillsAssignmentName|JobCode|SetCode|PersonNumber
MERGE|Assignee|ORA_CORE_JOB|AssignProgrammer_Python_Mongo|APPS_DEV_1|US_SET|3001005
MERGE|Assignee|ORA_CORE_JOB|AssignProgrammer_Python_Mongo|APPS_DEV_2|US_SET|3001005

METADATA|CoreSkill|Skill|SkillsAssignmentName|RequiredProficiencyLevelCode
MERGE|CoreSkill|Python|AssignProgrammer_Python_Mongo|3
MERGE|CoreSkill|MongoDB|AssignProgrammer_Python_Mongo|4
```

Example of Unassigning Skills

This example unassigns JAVA as a core skill for person 3001005.

```
METADATA|SkillsAssignment|SkillsAssignmentName|Action|AssignedByPersonNumber
MERGE|SkillsAssignment|UnAssign_Java|ORA_UNASSIGN|3001000

METADATA|Assignee|AssigneeType|SkillsAssignmentName|PersonNumber
MERGE|Assignee|ORA_CORE_PERSON|UnAssign_Java|3001005
METADATA|CoreSkill|Skill|SkillsAssignmentName
MERGE|CoreSkill|JAVA|UnAssign_Java
```

Guidelines for Loading Content Items

Upload content items to item catalogs using HCM Spreadsheet Data Loader (HSDL) or HCM Data Loader (HDL) using spreadsheet templates. These templates are designed to cater to the validations of the content type for which items are getting added or updated.

The following templates are delivered for each content type and can be used to create or update content items:

Template Code	Template Name
ORA_CONTENT_ITEM_PROJECTS	Merge Content Items - Projects
ORA_CONTENT_ITEM_BEHAVIORS	Merge Content Items - Behaviors
ORA_CONTENT_ITEM_ESTABLISHMENTS	Merge Content Items - Establishment

Template Code	Template Name
ORA_CONTENT_ITEM_HONORS	Merge Content Items - Honors and Awards
ORA_CONTENT_ITEM_LANGUAGES	Merge Content Items - Languages
ORA_CONTENT_ITEM_COMPETENCIES	Merge Content Items - Competencies
ORA_CONTENT_ITEM_DEGREES	Merge Content Items - Degrees
ORA_CONTENT_ITEM_EDUCATION_LEVELS	Merge Content Items - Education Levels
ORA_CONTENT_ITEM_CERTIFICATIONS	Merge Content Items - Licenses and Certifications
ORA_CONTENT_ITEM_MEMBERSHIPS	Merge Content Items - Memberships

Use the Spreadsheet Templates task in the Data Exchange work area to review these templates. Here you can make copies to customize or grant access to the roles that can use each template. The templates support create and update of content items.

Content item rating levels can be uploaded for the content types that support rating model. The following templates can be used to upload content item rating levels for specific content type:

- Merge Content Items - Behaviors
- Merge Content Items - Competencies
- Merge Content Items - Languages

In the above mentioned 3 templates, 10 instances of content item rating level are exposed. These can be used to override the rating level descriptions associated with the rating model for a given content item. If there are more than 10 rating levels, copy and create a new template and add the required number of instances of the content item rating level.

Related Topics

- Add a Role to a Predefined Spreadsheet Template

Guidelines for Loading Classic Talent Profile Data

The talent profile objects are Talent Profile, Content Item, Content Items Relationship, Education Establishment, and Rating Model. This topic describes aspects of the talent profile objects that you must understand to load them successfully using HCM Data Loader.

> **Note:** If you are still using Classic Talent Profiles, you should upgrade to Enhanced Talent Profiles before release 21A. For Person Profiles, see this file Enhanced Talent Profiles Section Template.pdf on My Oracle Support: Upgrading Oracle Cloud Profile Management (Doc ID 2421964.1). It describes the structure of each template that can be used to help create the TalentProfile.dat file to load Person Profile information via HCM Data Loader. Any relevant topics from this chapter have been updated and moved to My Oracle Support: Upgrading Oracle Cloud Profile Management (Doc ID 2421964.1).

Loading Talent Profiles

Talent profiles include person profiles and model profiles. A person profile contains information about a worker's skills, qualifications, accomplishments, and career preferences. A model profile contains information about the skills and qualifications required for a job or position. You can load both person and model profiles using HCM Data Loader.

The following table summarizes key information about some components of the Talent Profile object.

Component	Description
Profile Item	Holds the data for a content item. Content items are skills, qualifications, and qualities belonging to content types in the content library. For example, Communication Skills may be a content item in the Competencies content type. Profile Item components must be unique in the profile. Note that: • Duplicate profile items have the same qualifiers and overlapping start and end dates. • In case of person profile competency both QualifierId1 and QualifierId2 are mandatory. QualifierId1 specifies who added the competency, for example, the manager, HR, or self. QualifierId2 specifies who reviewed the change.
Profile Keyword	Holds keywords for areas of expertise and areas of interest in person profiles. Not valid for model profiles.
Profile Relation	Links a model profile to a job or position. These rules apply: • You can't link a model profile to a job family or organization when you upload the profile using HCM Data Loader. Any model profile that you load for those structures is therefore for information purposes only. • You can't associate a job or position more than once with the same profile. Not valid for person profiles.

A person can have only one person profile. You must supply a reference to the person for whom you're creating the person profile. You must not supply a person reference for a model profile.

> **Note:** The **Description** attribute of the Talent Profile object isn't required. However, you're recommended always to provide it for a person profile and include the worker name. Otherwise, the person profile doesn't appear in search results when you perform a basic search on the Search: Profiles page.

Loading Content Items

A content item is a specific occurrence of a content type. The associated content types and their field properties must exist in the target environment before you load content items to the content library. The field properties of the content type represent the information that you can capture for the content item.

Loading Content Items Relationships

A content items relationship defines the relationship between two content items. A content item can be either the parent or the child of another content item. These rules apply:

- The two content items must belong to different content types.

- You can't create more than one relationship between the same two content items. For example, content item A can't be both the parent and the child of content item B.
- You can't create a relationship between a content item and itself.
- The start and end dates of the content items relationship must not be outside the date ranges of the associated content items.
- The relationship between the associated content types must exist in the target environment before you load the related content items.

Loading Education Establishments

An education establishment is a school, college, university, or other organization that workers select when they add education information to their profiles. You may obtain a list of education establishments from a third party, for example. You can upload this information using HCM Data Loader.

These geography structures must exist in the target environment before you load education establishments that refer to them:

- Country
- State or Province

Loading Rating Models

You use rating models to rate workers on their performance and proficiency in the skills and qualities in the person profile. You can also use rating models to specify target proficiency levels for items in a model profile. You must include the parent Rating Model component in the same file as any child Rating Level and Rating Category components that you're loading.

Related Topics

- Oracle Fusion Profile Management Components: How They Work Together

Examples of Loading Talent Review Meetings

Talent review meetings are the main component of the talent review process, which organizations use to assess strengths of employees and evaluate areas of risk. You can use HCM Data Loader to create and update Talent Review meetings.

Create a Talent Review Meeting

This example TalentReviewMeeting.dat file creates a Talent Review Meeting object. The meeting has Facilitator, Participant, Reviewee, Review Content, Talent Pool, and Succession Plan components. The meeting configuration also has an attribute for including matrix managers as reviewers, which is set to **Y** in this example file.

> **Note:** Source keys identify each record. You must use source keys if you plan to update the facilitator, participant, or reviewee records later. The user key for these records includes the person number. Therefore, you can't update the person number to refer to a different person if you're identifying the record by the user key alone. You're recommended always to use source keys when creating talent review meetings so that you can update the meetings as necessary.

```
COMMENT * Talent Review Meeting *
METADATA|TalentReviewMeeting|SourceSystemId|SourceSystemOwner|DataSubmitDate|MeetingDate|
MeetingInstructions|MeetingPurpose|MeetingStatusCode|MeetingSubmitStatusCode|MeetingTitle|DataValidityCode|
DashboardTemplateName|MeetingLeaderPersonNumber|IncludeMatrixMgmt
MERGE|TalentReviewMeeting|MrktMgr071018|VISION|2021/01/28|2021/12/08|Instructions for Meeting|Purpose of
 Meeting|NOT_STARTED|DRAFT|HDL Marketing Talent Review Meeting|ONE_MONTH|Standard Talent Review Meeting
 Template|8153756|Y
COMMENT * Prior Rating Date Range *
METADATA|PriorRatingDateRange|SourceSystemId|SourceSystemOwner|FromDate|ToDate|MeetingTitle
MERGE|PriorRatingDateRange|MrktMgr071018_PDR|VISION|2020/01/01|2020/08/04|HDL Marketing Talent Review
 Meeting
COMMENT * Facilitator *
METADATA|Facilitator|SourceSystemId|SourceSystemOwner|MeetingTitle|FacilitatorPersonNumber
MERGE|Facilitator|MrktMgr071018_Fac1|VISION|HDL Marketing Talent Review Meeting|8153756
COMMENT * Participant *
METADATA|Participant|SourceSystemId|SourceSystemOwner|DataSubmissionStatusCode|ParticipantTypeCode|
MeetingTitle|ParticipantPersonNumber
MERGE|Participant|MrktMgr071018_Part|VISION|NOT_STARTED|PARTICIPANT|HDL Marketing Talent Review Meeting|
8153780
COMMENT * Review Content *
METADATA|ReviewContent|SourceSystemId|SourceSystemOwner|UseAsAxisFlag|MeetingTitle|Rating
MERGE|ReviewContent|MrktMgr071018_Rate_Perf|VISION|Y|HDL Marketing Talent Review Meeting|Performance
MERGE|ReviewContent|MrktMgr071018_Rate_Pot|VISION|Y|HDL Marketing Talent Review Meeting|Potential
MERGE|ReviewContent|MrktMgr071018_Rate_Impct|VISION|Y|HDL Marketing Talent Review Meeting|Impact of Loss
MERGE|ReviewContent|MrktMgr071018_Rate_Risk|VISION|N|HDL Marketing Talent Review Meeting|Risk of Loss
MERGE|ReviewContent|MrktMgr071018_Rate_Comp|VISION|Y|HDL Marketing Talent Review Meeting|Overall
 Competencies Rating
MERGE|ReviewContent|MrktMgr071018_Rate_Goals|VISION|N|HDL Marketing Talent Review Meeting|Overall Goals
 Rating
MERGE|ReviewContent|MrktMgr071018_Rate_Talent|VISION|N|HDL Marketing Talent Review Meeting|Talent Score
COMMENT * Reviewees *
METADATA|Reviewee|SourceSystemId|SourceSystemOwner|MeetingTitle|RevieweePersonNumber
MERGE|Reviewee|MrktMgr071018_Rev1|VISION|HDL Marketing Talent Review Meeting|8153758
MERGE|Reviewee|MrktMgr071018_Rev1|VISION|HDL Marketing Talent Review Meeting|8153761
COMMENT * Succession Plan *
METADATA|SuccessionPlan|SourceSystemId|SourceSystemOwner|MeetingTitle|PlanName
MERGE|SuccessionPlan|MrktMgr071018_SP|VISION|HDL Marketing Talent Review Meeting|HDL_JOB_PLAN1
COMMENT * Talent Pool *
METADATA|TalentPool|SourceSystemId|SourceSystemOwner|MeetingTitle|PoolName|PoolStatus
MERGE|TalentPool|MrktMgr071018_TP|VISION|HDL Marketing Talent Review Meeting|HDL_TOP_POOL|Active
```

These rules apply to Talent Review Meeting objects that you create using HCM Data Loader:

- The **DashboardTemplateName** attribute identifies the configuration template used for the meeting. The template is created in the Setup and Maintenance work area using the **Configure Talent Review Dashboard Options** task. The referenced template must exist before you load the talent review meeting record.
- Any facilitator, participant, reviewee, talent pool, and succession plan records that are referenced by the talent review meeting record must already exist. Alternatively, you can include them in the same .zip file.
- You must identify at least one facilitator.
- The **MeetingStatusCode** attribute must be set to **NOT_STARTED** and the **MeetingSubmitStatusCode** attribute can be set to either **DRAFT** or **SCHEDULED** for new talent review meetings.

Update a Talent Review Meeting

The following table lists actions that you can perform when you update a talent review meeting using HCM Data Loader. Permitted actions depend on the meeting status.

Meeting Status	Permitted Actions
Not started	• Change the meeting template. • Change the meeting status to **In progress** or **Canceled**. • Add or remove reviewees, participants, and facilitators.
In progress	• Change the meeting status to **Not started** or **Canceled**. • Add or remove reviewees, participants, and facilitators.
Complete	Update the facilitator.
Canceled	Change the meeting status to **Not started**.
Process job in progress	None.
Process job error	None.

Update the Facilitator

This example TalentReviewMeeting.dat file updates the facilitator for an existing meeting. This example uses source keys. You can't use user keys alone to update the facilitator, as the **FacilitatorPersonNumber** attribute that identifies the new facilitator also identifies the record to update.

```
METADATA|Facilitator|SourceSystemId|SourceSystemOwner|MeetingTitle|FacilitatorPersonNumber
MMERGE|Facilitator|MrktMgr071018_Fac2|VISION|HDL Marketing Talent Review Meeting|17121468
```

Tip: If you're making multiple changes to the same talent review meeting, then include the Talent Review Meeting record in the same .dat file to ensure that all changes are processed together.

Remove Stakeholders

Sometimes workers may no longer be part of the organization. You may want to remove terminated workers who were added as facilitators and participants from Talent Review meetings.

This example TalentReviewMeeting.dat file is used for deleting a facilitator's and a participant's data from the HDL Marketing Talent Review Meeting meeting. This file uses source keys to identify the facilitator and participant to delete.

```
METADATA|Facilitator|SourceSystemId|SourceSystemOwner|MeetingTitle|FacilitatorPersonNumber
DELETE|Facilitator|MrktMgr071018_Fac2|VISION|HDL Marketing Talent Review Meeting|17121468
METADATA|Participant|SourceSystemId|SourceSystemOwner|MeetingTitle|ParticipantPersonNumber
DELETE|Participant|MrktMgr071018_Part1|VISION|HDL Marketing Talent Review Meeting|8153780
```

Related Topics

- Talent Review
- Talent Review Template

FAQs for Loading Talent Objects

How can I transfer two performance documents in HCM Data Loader from an employee's old assignment to the new assignment when the performance documents are associated with different review periods?

Create one file including both performance documents for the employee, but don't add the review period and then both the performance documents will be loaded. Or create a different file for each performance document and include the specific review period.

Can I load any type of requisition using the Requisition object?

No. You can load:

- Vacancies, which are placeholders for future headcount. Use vacancies if you don't have a dedicated recruiting application.
- Oracle Talent Acquisition Cloud requisitions.

You can't load Oracle Recruiting Cloud requisitions using the Requisition object. To load Oracle Recruiting Cloud requisitions, use the Job Requisition object.

How can I change public succession plans to private succession plans using HCM Data Loader

Use the **AccessTypeCode** attribute to change the status of a succession plan. Set its value to PRIVATE when uploading succession plan details using HCM Data Loader.

Here's an example SuccessionPlan.dat file for making a public plan BJ_Succession_Plan private.

```
METADATA|SuccessionPlan|PlanName|PlanType|Status|AccessTypeCode
MERGE|SuccessionPlan|BJ_Succession_Plan|INCUMBENT|ACTIVE|PRIVATE
```

36 Loading Questionnaire Objects

Overview of Loading Questions, Questionnaire Templates, and Questionnaires

You can load questions, questionnaire templates, and questionnaires using HCM Data Loader. Load the objects together or independently, but ensure that you provide the relevant information.

If you provide all these objects together, HCM Data Loader loads them in the correct order. These objects don't support integration keys, instead supply the user keys.

Business Objects	Components
Question	Question Answer
Questionnaire Template	Questionnaire Template Questionnaire Template Section Questionnaire Template Question Questionnaire Template Question Answer Questionnaire Template Attachment
Questionnaire	Questionnaire Questionnaire Section Questionnaire Question Questionnaire Answer Questionnaire Attachment

All the Questionnaire objects such as questionnaire template, questionnaire, and question supports only user keys and not any other.

Guidelines for Loading Questions, Questionnaire Templates, and Questionnaires

Here are a few guidelines when using HCM Data Loader for loading Questionnaires data:

- Supports the Create, Update, and Delete operations, including translated objects.
- Identifies if a row is created or updated with the help of user keys. Only user keys are supported and not the source keys or surrogate keys. For example, if the Question Version Number in the data file is the same as the Question Version Number in the application, then the question is updated.
- For the Recruiting subscriber, you can associate questions with additional contexts such as Job Family, Job Function, Organization, and Location.

Updating Questionnaire Data

When updating a Questionnaire, you must supply the complete Questionnaire object in the Questionnaire.dat file, and not just the changed records. The parent record is required while updating any of the child records. The Questions and Answers can't be deleted using the Questionnaire.dat file, instead use the Question.dat file.

Deleting Questionnaire Data

You can delete Questionnaire objects using HCM Data Loader if you identify their user keys.

- When you delete individual child components of the object, the parent record must be included in the same data file.
- When you delete a Questionnaire object, the child components such as sections, questions, and answers are also deleted.
- When you delete the parent Questionnaire Template object, you must delete Questionnaire components explicitly.

Related Topics

- Examples of Associating a Question with a Subscriber Context

Examples of Loading Questions

This topic provides an example showing how to load a multiple-choice question with 3 answer choices for the subscriber, Recruiting. Before creating a question, be sure that the category it's assigned to exists.

You can create the category as a subfolder of the Question Library for the subscriber by using the Question Library task in Setup and Maintenance work area, Workforce Development offering.

Question

This example Question.dat file shows how to load a multiple-choice question with 3 answer choices for the Recruiting subscriber using the user keys.

```
METADATA|Question|QuestionCode|QstnVersionNum|QuestionText|SubscriberName|Status|QuestionType|PrivacyFlag|
ResponseTypeName|CategoryName
MERGE|Question|FEEDBACK_1|1|Are you satisfied with the training?|Recruiting|A|MULTCHOICE|N|Check Multiple
 Choices|Feedback
METADATA|Answer|QuestionCode|AnswerCode|QstnVersionNum|LongText|SequenceNumber|Score|SubscriberName
MERGE|Answer|FEEDBACK_1|FEEDBACK_1ANS11|1|Yes, I'm satisfied.|5|151|Recruiting
MERGE|Answer|FEEDBACK_1|FEEDBACK_1ANS21|1|No, I'm not satisfied.|3|151|Recruiting
MERGE|Answer|FEEDBACK_1|FEEDBACK_1ANS31|1|I have no opinion on this.|2|151|Recruiting
```

Examples of Associating a Question with a Subscriber Context

You can associate contexts to questions that are loaded to the application using HCM Data Loader for the Oracle Recruiting subscriber only.

Scenario

When you create questions in the question library, you can associate the question to one or more of these dimensions:

- Job Families
- Recruiting Organizations
- Job Functions
- Locations

By default the questions applies to all the dimensions as shown by the fields, **Recruiting Organizations**, **Recruiting Locations**, **Job Families** and **Job Functions**.

You can restrict a question to a specific value in a dimension. For example, you can specify a Job Location such as New York. In that case, set the field to **N**, and the specific dimension gets attached to the question.

In HCM Data Loader, the business object Question Context is a child object of Question. The Question Context business object has 4 user keys:

1. QuestionCode, QstnVersionNum, LocationName
2. QuestionCode, QstnVersionNum, JobFamilyName
3. QuestionCode, QstnVersionNum, OrganizationName
4. QuestionCode, QstnVersionNum, JobFunction

Each set represents a dimension.

Note these guidelines:

- The parent question must always be in the dat file because Question Context isn't integration enabled.
- Question Context applies to the Recruiting subscriber only. For other subscribers, the load fails.
- Question Context business objects support only user keys. SurrogateId load isn't supported.

- When you load a Question Context to a Question, the MERGE line can have only 1 dimension.
- For the dimension loaded in the Question Context, the corresponding "All%Flag" should be **N**.
- If the "All%Flag" for a dimension is **N**, there should at least 1 Question Context in that dimension.

To use HCM Spreadsheet Data Loader, go to the Data Exchange work area, create a spreadsheet template and then upload it using the **Load Spreadsheet Data** task.

Sample Question File

Tip: You can load only one context per line in the question.dat input data file.

Question 1 in Question.dat associates a question to a dimension in Oracle Recruiting.

Questions 2 and 3 fail. Question 2 fails because JobFamilyName and LocationName are on the same line. Question 3 fails because AllJobFamiliesFlag isn't **N**.

```
METADATA|Question|QuestionCode|QstnVersionNum|SubscriberName|QuestionType|ResponseTypeName|CategoryName|
ClassificationCode|QuestionText|CandidateCode|AllJobFamiliesFlag|AllOrganizationsFlag|AllJobFunctionsFlag|
AllLocationsFlag
METADATA|Answer|QstnVersionNum|QuestionCode|SequenceNumber|AnswerCode|LongText|SubscriberName|
AddAttachmentFlag|AttachmentTitle|AttachmentDescription|AttachmentFile|AttachmentFileName|Type
METADATA|QuestionContext|JobFamilyName|LocationName|JobFunction|OrganizationName|QuestionContextId|
QstnVersionNum|QuestionCode|SubscriberName
COMMENT ------------Question 1----------------------------------------
MERGE|Question|RECRUIT_INTERVIEW_1|1|Recruiting|||Question Library||Is the Candidate willing to work in
 shifts?|BOTH|N|N|N|N
MERGE|Answer|1|RECRUIT_INTERVIEW_1|234|RECRUIT_INTERVIEW_ANS1|Yes|Recruiting||||||
MERGE|Answer|1|RECRUIT_INTERVIEW_1|235|RECRUIT_INTERVIEW_ANS2|No|Recruiting||||||
MERGE|QuestionContext|ZFRCE_Planning|||||1|RECRUIT_INTERVIEW_1|Recruiting
MERGE|QuestionContext||Germany||||1|RECRUIT_INTERVIEW_1|Recruiting
MERGE|QuestionContext|||Legal|||1|RECRUIT_INTERVIEW_1|Recruiting
MERGE|QuestionContext||||ZBEN United Benefits Corporation||1|RECRUIT_INTERVIEW_1|Recruiting
COMMENT ------------Question 2----------------------------------------
MERGE|Question|RECRUIT_INTERVIEW_2|1|Recruiting|||Question Library||Is the Candidate ready to travel as part
 of Job?|BOTH|N|N|N|N
MERGE|Answer|1|RECRUIT_INTERVIEW_2|234|RECRUIT_INTERVIEW_2_ANS1|Yes|Recruiting||||||
MERGE|Answer|1|RECRUIT_INTERVIEW_2|235|RECRUIT_INTERVIEW_2_ANS2|No|Recruiting||||||
MERGE|QuestionContext|ZFRCE_Planning|Germany||||1|RECRUIT_INTERVIEW_2|Recruiting
COMMENT ------------Question 3----------------------------------------
MERGE|Question|RECRUIT_INTERVIEW_3|1|Recruiting|||Question Library||Is the candidate ready to migrate as
 part of Job?|BOTH|||N|N
MERGE|Answer|1|RECRUIT_INTERVIEW_3|234|RECRUIT_INTERVIEW_3_ANS1|Yes|Recruiting||||||
MERGE|Answer|1|RECRUIT_INTERVIEW_3|235|RECRUIT_INTERVIEW_3_ANS2|No|Recruiting||||||
MERGE|QuestionContext|ZFRCE_Planning|||||1|RECRUIT_INTERVIEW_3|Recruiting
MERGE|QuestionContext||Germany||||1|RECRUIT_INTERVIEW_3|Recruiting
MERGE|QuestionContext|||Legal|||1|RECRUIT_INTERVIEW_3|Recruiting
MERGE|QuestionContext||||ZBEN United Benefits Corporation||1|RECRUIT_INTERVIEW_3|Recruiting
```

To delete the Question Context, change the MERGE to DELETE.

Examples of Loading Questionnaire Templates

This topic provides an example to show how to load a questionnaire template including the sections, questions, and responses using HCM Data Loader for the subscriber, Recruiting.

Questionnaire Template

This example shows the QuestionnaireTemplate.dat file to load questionnaire template components using the user keys.

- A questionnaire template is created with 2 sections, 606013 and 60614. Questions and answer choices are provided for the section 606013.
- The **AllJobFamiliesFlag**, **AllJobFunctionsFlag**, **AllLocationsFlag** and **AllOrganizationsFlag** will default to **Y** because the HCM Data Loader doesn't support associating a question to a specific job family, job function, location, or organization.

```
METADATA|QuestionnaireTemplate|SubscriberName|QuestionnaireCode|Name|Status|AllowIntroChg|AllowFormatChg|
AllowSectionChg|AllowQstnOrderChg|AllowRespOrderChg|SectionOrder|SectionPresentation|PrivacyFlag|
PlainTextResponse|SingleChoiceListResponse|RichTextResponse|MultipleChoiceListResponse|NoneResponse|
SingleChoiceRadioResponse|MultipleChoiceCheckResponse|OwnerPersonNumber
MERGE|QuestionnaireTemplate|Recruiting|RECRUIT_QSTNR_TMPL_1|RECRUIT_QSTNR_TMPL_1|A|Y|N|Y|Y|Y|R||N|Y|N|N|N||
Y||180807
METADATA|Section|SubscriberName|QuestionnaireCode|SectionSeqNum|Name|NewPage|Description|IntroText|
QuestionOrder|ResponseOrder|Mandatory|AllowAdhoc
MERGE|Section|Recruiting|RECRUIT_QSTNR_TMPL_1|606013|RECRUIT_QSTNR_TMPL_1_SEC_1|Y|Introduction|Section for
 Introduction|V|R|Y|Y
MERGE|Section|Recruiting|RECRUIT_QSTNR_TMPL_1|606014|RECRUIT_QSTNR_TMPL_1_SEC_2|Y|Feedback|Section for feed
 back questions|R|R|N|N
METADATA|Question|SubscriberName|QuestionnaireCode|SectionSeqNum|QuestionCode|QstnVersionNum|SeqNum|
KeepWithPrev|AdhocQstn|Mandatory|ResponseTypeName|Status|QuestionText|CategoryName|QuestionType|PrivacyFlag|
RatingModelCode|ConditionDisplay|ConditionQstnVersionNum|ConditionQuestionCode|ConditionAnswerCode|
AllJobFamiliesFlag|AllJobFunctionsFlag|AllLocationsFlag|AllOrganizationsFlag|CandidateCode|
ClassificationCode|QstnResponseTypeName|AdhocFlag
MERGE|Question|Recruiting|RECRUIT_QSTNR_TMPL_1|606013|RECRUIT_QSTN_EXPERTISE|1|32956||N|Y|Multiple Choices
 from List|D|What is your area of expertise?|Question Library|MULTCHOICE|N|TALENTSCORE|||||Y|Y|Y|Y|BOTH|
DISQUALIFICATION|Multiple Choices from List|
METADATA|Answer|SubscriberName|QuestionnaireCode|SectionSeqNum|QuestionCode|QstnVersionNum|SeqNum|
AnswerCode|ShortText|LongText|Score
MERGE|Answer|Recruiting|RECRUIT_QSTNR_TMPL_1|606013|RECRUIT_QSTN_EXPERTISE|1|329456|
RECRUIT_QSTN_EXPERTISE_1||Java|
MERGE|Answer|Recruiting|RECRUIT_QSTNR_TMPL_1|606013|RECRUIT_QSTN_EXPERTISE|1|329457|
RECRUIT_QSTN_EXPERTISE_2||PL/SQL|
MERGE|Answer|Recruiting|RECRUIT_QSTNR_TMPL_1|606013|RECRUIT_QSTN_EXPERTISE|1|329458|
RECRUIT_QSTN_EXPERTISE_3||Python|
```

Examples of Loading Questionnaires

This topic provides examples showing how to load a questionnaire including the sections, questions, responses, and attachments using HCM Data Loader for the subscriber, Recruiting.

Questionnaire

In this example the Questionnaire.dat file creates a questionnaire using user keys.

- The questionnaire has one section with one question, and three answer choices.
- The question created for the questionnaire isn't in the question library.
- The answers include attachments. Attachment files must be provided in the BlobFiles subdirectory of same the compressed .zip file that you supply Questionnaire.dat file in.

```
METADATA|Questionnaire|QuestionnaireCode|QstnrVersionNum|QuestionnaireName|Status|SubscriberName|
PrivacyFlag|CategoryName|UpdateAllowed|SectionOrder|SectionPresentation|PageLayout|InUse|LatestVersion|
TemplateQstnrCode
MERGE|Questionnaire|NVVF1|1|NV_HDL_QSTNR_1009_1|Active|Recruiting|Public|Questionnaire Library|Y|Sequential|
Stack Regions|1|No|Y|NV_HDL_QSTNR_1009
METADATA|Section|SectionSeqNum|QuestionnaireCode|QstnrVersionNum|NewPage|SubscriberName|IntroText|Name
MERGE|Section|1|NVVF1|1|Y|Recruiting|A sample section.|Training satisfaction
METADATA|Question|QuestionCode|QstnVersionNum|SectionSeqNum|QuestionnaireCode|QstnrVersionNum|QuestionText|
SubscriberName|Status|QuestionType|PrivacyFlag|AdhocQstn|CategoryName|QstnSequenceNumber|ResponseTypeName|
ScoredFlag|MaxPossibleScore|MinThresholdScore|AddAttachmentFlag|AttachmentTitle|AttachmentDescription|
AttachmentFile|AttachmentFileName
MERGE|Question|NVVF1|1|1|NVVF1|1|How do you rate the HDL Training?|Recruiting|A|Multiple Choice|Public|N|
Question Library|5|Check Box with Image|Y|10|6|Y|Introduction||Text.txt|Text
METADATA|Answer|QuestionCode|QstnVersionNum|AnswerCode|LongText|AnsSequenceNumber|Score|SectionSeqNum|
QuestionnaireCode|QstnrVersionNum|SubscriberName|AddAttachmentFlag|AttachmentTitle|AttachmentDescription|
AttachmentFile|AttachmentFileName|Type
MERGE|Answer|NVVF1|1|NVVF1Answer14|Average|1|8|1|NVVF1|1|Recruiting|Y|Intro||Text.txt|Text|FILE
MERGE|Answer|NVVF1|1|NVVF1Answer16|Very Good|1|9|1|NVVF1|1|Recruiting|Y|Intro||Text.txt|Text|FILE
MERGE|Answer|NVVF1|1|NVVF1Answer18|Excellent|1|7|1|NVVF1|1|Recruiting|Y|extract_runs||extract_runs.xml|
extract_runs|FILE
```

Questionnaire with Attachments

This example shows how to load a questionnaire including the attachments for the questionnaire, question, and response using HCM Data Loader for the subscriber, Recruiting. In this example, the Questionnaire.dat file creates a questionnaire with attachments for the questionnaire, question, and response. The user keys are used to identify each record.

- The questionnaire has an attachment at the questionnaire-level, question-level, and response-level.
- The answers include attachments. Attachment files must be provided in the BlobFiles subdirectory of same the compressed zipped file that you supply Questionnaire.dat file in.

```
METADATA|Questionnaire|QuestionnaireCode|QstnrVersionNum|QuestionnaireName|Status|SubscriberName|
PrivacyFlag|CategoryName|UpdateAllowed|SectionOrder|SectionPresentation|PageLayout|InUse|LatestVersion|
TemplateQstnrCode
METADATA|QuestionnaireAttachment|AttachmentId|Title|QuestionnaireCode|QstnrVersionNum|SubscriberName|
QuestionnaireId|File|FileName|Type
METADATA|Question|QuestionCode|QstnVersionNum|SectionSeqNum|QuestionnaireCode|QstnrVersionNum|
QuestionText|SubscriberName|Status|QuestionType|PrivacyFlag|AdhocQstn|CategoryName|QstnSequenceNumber|
QstnrQstnResponseTypeName|ScoredFlag|MaxPossibleScore|MinThresholdScore|AddAttachmentFlag|AttachmentTitle|
AttachmentDescription|AttachmentFile|AttachmentFileName|DeleteAttachmentFlag
METADATA|Answer|QuestionCode|QstnVersionNum|AnswerCode|LongText|AnsSequenceNumber|Score|SectionSeqNum|
QuestionnaireCode|QstnrVersionNum|SubscriberName|AddAttachmentFlag|AttachmentTitle|AttachmentDescription|
AttachmentFile|AttachmentFileName|Type
METADATA|Section|SectionSeqNum|QuestionnaireCode|QstnrVersionNum|NewPage|SubscriberName|IntroText|Name
MERGE|Questionnaire|QSTNR_HDL_100|1|Interview Feedback|Active|Recruiting|Public|Questionnaire Library|Yes|
Sequential|Stack Regions|1|No|Y|HS_QSTNR_10
MERGE|QuestionnaireAttachment||Instructions|QSTNR_HDL_100|1|Recruiting||Instructions.txt|Instructions|FILE
MERGE|Section|1|QSTNR_HDL_100|1|Yes|Recruiting|Give your valuable feedback|Feedback Section
MERGE|Question|QSTNR_100_QSTN_1|1|1|QSTNR_HDL_100|1|How was the interview experience?|Recruiting|A|
Multiple Choice|Public|N|Question Library|5|Check Box with Image|Y|5|1|Y|Introduction||Introduction.png|
Introduction|
MERGE|Answer|QSTNR_100_QSTN_1|1|QSTNR_100_QSTN_1_ANS_1|Excellent|1|5|1|QSTNR_HDL_100|1|Recruiting|Y|Option||
Option1.png|Option|FILE
MERGE|Answer|QSTNR_100_QSTN_1|1|QSTNR_100_QSTN_1_ANS_2|Good|1|3|1|QSTNR_HDL_100|1|Recruiting|Y|Option||
Option2.png|Option|FILE
MERGE|Answer|QSTNR_100_QSTN_1|1|QSTNR_100_QSTN_1_ANS_3|Poor|1|1|1|QSTNR_HDL_100|1|Recruiting|Y|Option||
Option3.png|Option|FILE
```

Related Topics

- How You Load Images, Attachments, and Large Strings

37 Loading Courses and Offerings to the Learning Catalog

Guidelines for Loading a Course to Oracle Learning Using CourseV3

If you have activity sequencing turned on, you can load courses, offerings, activities, and classroom and instructor reservations to the Oracle Learning catalog using the CourseV3 object. If it's turned off, use the Course object.

> **Tip:** Unused Attributes can get deprecated, so always use the most recent DAT file.

To verify that the course successfully loaded, use the Courses task on the **My Client Groups** > **Learning** page. To verify that an offering, activities, and classroom and instructor reservations successfully loaded, use the Offerings task.

Here's the hierarchy and processing order for the relevant learning objects:

1. CourseV3
    a. CourseDefaultAccessV3
    b. OfferingV3
        i. OfferingDefaultAccessV3
        ii. EvaluationActivityV3
        iii. OfferingActivitySectionV3
            a. SelfPacedActivityV3
            b. InstructorLedActivityV3
                1. AdhocResourceV3
                2. ClassRoomReservationV3
                3. InstructorReservationV3
2. CourseV3Translation
3. OfferingV3Translation
4. SelfPacedActivityV3Translation
5. InstructorLedActivityV3Translation
6. OfferingActivtiySectionV3Translation
7. EvaluationActivityV3Translation

When you create the course hierarchy, you need to create the sections and activities in a separate .dat file from the course and offerings. Here's how you create the complete course hierarchy:

1. Load the CourseV3 and OfferingV3 objects using a single CourseV3.dat file.
    - Make sure that the required CourseID, CourseNumber, EffectiveStartDate, OwnedByPersonId, and Title attributes all have the appropriate values.
    - Make sure that the OfferingNumber doesn't exceed 26 characters. Otherwise the autogenerated SectionNumber for the default section will exceed 30 characters and result in an HCM Data Loader error.

2. Load the EvaluationActivityV3, OfferingActivitySectionV3, SelfPacedActivityV3, InstructorLedActivityV3, and other applicable objects together in a separate CourseV3.dat file.
    - Make sure that the .dat file also includes the parent OfferingV3 reference, even if you aren't updating the parent. Otherwise the load process fails with an error.
    - For all activities, use the OfferingSectionId and OfferingSectionNumber values from the OfferingActivitySectionV3 object as the parent surrogate ID and user key respectively.

Rich Text Syllabus and Cover Art

A course file can have a rich text syllabus description and cover art.

- To load rich text, you need to reference a rich text file in the Syllabus attribute, such as an .html. To load the rich text files, place them all in a folder called ClobFiles and zip the folder. Then include that zip inside the Course.zip file.
- To load cover art, you need to reference an image file in the CoverArtFile and CoverArtFileName attributes, such as a .jpg file. To load the image files, place them in a folder called BlobFiles and zip the folder. Then include that zip inside the Course.zip file.

Course Validations

The load process does these validations:

- On create, the CourseNumber value doesn't exist and the Title and OwnedByPersonID values aren't null
- The Person object exists for the provided OwnedByPersonId value
- The Trailer Video record for the provided TrailerLiId value exists in Oracle Learning
- Checks if a CoverArtFile is provided and if so, that the CoverArtFile extension is one of these valid file formats: .jpeg, .jpg, .bmp, .tiff, .png, or .gif, and can be read
- The EffectiveStartDate value isn't empty or null
- The PublishStartDate value is before the PublishEndDate value
- The MinimumExpectedEffort value is either 0 or a positive number
- The MaximumExpectedEffort, MinimumPrice, and MaximumPrice values are a positive number
- The MinimumExpectedEffort value is less than the MaximumExpectedEffort value
- The MinimumPrice value is less than the MaximumPrice value
- It can retrieve the access properties for the learning item

Offering Validations

The load process does these validations on create:

- The PersonId value isn't null and the Person and InstructorResource objects already exist for the provided PersonId value
- The InstructorResourceNumber value exists in Oracle Learning
- The OwnedByPersonId value isn't null and the Person object already exists for the provided OwnedByPersonId value
- The OfferingNumber value doesn't already exist in Oracle Learning
- The Title, OfferingType, and CourseId values aren't null

It also does these validations:

- The values for these attributes aren't empty or null:
  - EffectiveStartDate
  - OfferingStartDate
  - OfferingEndDate
  - MaximumCapacity when the EnableCapacity value is 'Y'
  - PrimaryLocation ILT and blended offerings
    - For virtual classrooms, set the PrimaryLocationId value -1
    - For not defined classrooms, set the PrimaryLocationId value -2
  - Coordinator
  - FacilitatorType when you provide either a PrimaryInstructorId or TrainingSupplierId value
  - TrainingSupplierId when the FacilitatorType value is ORA_TRNG_VENDOR
  - PrimaryInstructorId when FacilitatorType is ORA_INSTRUCTOR
- The Person object exists for the provided PersonID or OwnedByPersonId values
- The course details for the provide CourseLearningItemId value exist in Oracle Learning
- The Questionnaire record for the provided QuestionnaireCode value exists in Oracle Learning
- A default questionnaire for the provided OfferingId value doesn't exist
- The PublishStartDate value is before the PublishEndDate value
- The offering PublishStartDate value is after the course PublishStartDate value
- The MinimumCapacity value is 0 or a positive whole number
- The MaximumCapacity value is a positive whole number
- The MinimumCapacity value is less than the MaximumCapacity value
- The EnableCapacity and EnableWaitList values are either 'Y' and 'N'
- The AccessPermissionId and TrainingSupplierId values exist in Oracle Learning
- The FacilitatorType value is ORA_TRNG_VENDOR or ORA_INSTRUCTOR
- An offering has only one evaluation activity otherwise the process ends in error
  - To use the default evaluation, for QuestionnaireCode, enter -1. To use a specific evaluation, enter its questionnaire code.

Default Section

The load process automatically does this work for the default section of an offering:

- Sets SectionTitle to Default Section.
- Automatically generates the SectionNumber value using the parent OfferingNumber and the _SEC suffix. For example, if the offering number is OFFERING-202104280937-HDL , then the generated section number is OFFERING-202104280937-HDL_SEC.
- Sets Position to 1. If you add more sections to the offering, set the Position values in the appropriate sequence, starting at 2.

- Sets ShowSectionIfSingleSectionExists to N so that the section isn't visible to learners. If you add more sections to the offering, the load process ignores any value for this attribute. This attribute is equivalent to the Enable as a visible section to learners option on the Activities tab of the offering details page.

CourseV3 DAT File Structure with Attributes

```
METADATA|CourseV3|CourseId|EffectiveStartDate|EffectiveEndDate|CourseNumber|Title|ShortDescription|Syllabus|
PublishStartDate|PublishEndDate|MinimumExpectedEffort|MaximumExpectedEffort|CurrencyCode|MinimumPrice|
MaximumPrice|CoverArtFile|CoverArtFileName|OwnedByPersonId|OwnedByPersonNumber|SourceType|SourceId|
SourceInfo|SourceSystemOwner|SourceSystemId|GUID|SyllabusText

METADATA|CourseDefaultAccessV3|DefaultAccessId|DefaultAccessNumber|EffectiveStartDate|EffectiveEndDate|
FollowCommunity|FollowSpecialization|SelfViewMode|SelfInitialStatus|SelfEnrollForm|SelfActivateApprove|
SelfEnrollQuestionnaire|SelfQuestionnaireCode|SelfPrereqType|SelfWithdrawPrereqDays|MgrInitialStatus|
MgrActivateApprove|MgrWithdrawPrereqDays|MgrMarkComplete|SourceSystemId|SourceSystemOwner|GUID|CourseId|
CourseNumber

METADATA|OfferingV3|OfferingId|EffectiveStartDate|EffectiveEndDate|OfferingNumber|OwnedByPersonId|SourceId|
SourceInfo|SourceType|SourceSystemOwner|SourceSystemId|GUID|LanguageCode|Title|Description|MaximumCapacity|
MinimumCapacity|OfferingType|CourseId|CourseNumber|EnableCapacity|EnableWaitlist|PrimaryLocationId|
PrimaryInstructorId|TrainingSupplierId|FacilitatorType|CoordinatorId|CoordinatorNumber|DescriptionText|
OfferingStartDate|OfferingEndDate|OwnedByPersonNumber|PrimaryInstructorNumber|PrimaryLocationNumber|
PublishStartDate|PublishEndDate|TrainingSupplierNumber|ShowSectionIfSingleSectionExists

METADATA|EvaluationActivityV3|ActivityId|EffectiveStartDate|EffectiveEndDate|ActivityNumber|Title|
OfferingId|OfferingNumber|QuestionnaireCode|QstnrRequiredForCompletion|SourceSystemId|SourceSystemOwner|GUID

METADATA|OfferingActivitySectionV3|OfferingSectionId|EffectiveStartDate|EffectiveEndDate|
OfferingSectionNumber|SourceSystemOwner|SourceSystemId|GUID|Title|Description|NumberOfActivitiesToComplete|
CompletionRuleType|OfferingId|SectionPosition|SequenceRuleType|OfferingNumber|DescriptionText|
SequenceRuleSectionNumber

METADATA|InstructorLedActivityV3|ActivityId|EffectiveStartDate|EffectiveEndDate|ActivityNumber|
ExpectedEffort|SourceInfo|SourceType|SourceId|SourceSystemOwner|SourceSystemId|GUID|Title|ShortDescription|
Description|CompletionRuleType|VirtualClassroomUrl|SelfCompleteFlag|TimeZone|OfferingSectionId|
ActivityPosition|SequenceRuleType|DescriptionText|OfferingSectionNumber|ActivityDate|ActivityStartTime|
ActivityEndTime|SequenceRuleActivityNumber|ClassroomResourceId|ClassroomResourceNumber|VirtualProviderId|
VirtualProviderNumber|VirtualProviderProduct

METADATA|AdhocResourceV3|AdhocResourceId|AdhocResourceNumber|Title|Description|Quantity|ActivityId|
ActivityNumber|SourceSystemId|SourceSystemOwner|GUID

METADATA|ClassroomReservationV3|ClassroomReservationId|ClassroomReservationNumber|ClassroomResourceId|
ClassroomResourceNumber|ActivityId|ActivityNumber|SourceSystemId|SourceSystemOwner|GUID

METADATA|InstructorReservationV3|InstructorReservationId|InstructorReservationNumber|InstructorResourceId|
InstructorResourceNumber|ActivityId|ActivityNumber|SourceSystemId|SourceSystemOwner|GUID

METADATA|SelfPacedActivityV3|ActivityId|EffectiveStartDate|EffectiveEndDate|ActivityNumber|ExpectedEffort|
SourceInfo|SourceType|SourceId|SourceSystemOwner|SourceSystemId|GUID|Title|ShortDescription|Description|
ContentId|CompletionRuleType|SelfCompleteFlag|OfferingSectionId|ActivityPosition|SequenceRuleType|
OfferingSectionNumber|DescriptionText|ContentNumber|SequenceRuleActivityNumber

METADATA|OfferingDefaultAccessV3|DefaultAccessId|DefaultAccessNumber|EffectiveStartDate|EffectiveEndDate|
SelfViewMode|SelfInitialStatus|SelfEnrollForm|SelfActivateApprove|SelfEnrollQuestionnaire|
SelfQuestionnaireCode|MgrInitialStatus|MgrActivateApprove|MgrWithdrawPrereqDays|SourceSystemId|
SourceSystemOwner|GUID|OfferingId|OfferingNumber
```

How Completion and Sequencing Rules Map to Learning Section Attributes in CourseV3

These OfferingV3 attributes support section-level completion and sequencing rules in the CourseV3.dat file.

Attribute	Description	Valid Values	Business Rules
CompletionRuleType	Determines whether the completion rule is defined at the Section or Activity level	ORA_COMPLETION_ACTIVITY or ORA_COMPLETION_SECTION	When you use ORA_COMPLETION_SECTION, you need to provide a valid NumberOfActivitiesToComplete value.
NumberOfActivitiesToComplete	Number of activities a learner needs to complete, to complete the section when CompletionRuleType is Section	When all section activities are required: -1 Otherwise a numeric value between 0 and n where n is the total number of section activities.	The specified value can't exceed the total number of activities. When CompletionRuleType is ORA_COMPLETION_ACTIVITY, the load process ignores any value that you provide.
SequenceRuleType	Determines when a learner can access the section	ORA_SECTION_ANYTIME: Learner can access the section anytime ORA_SECTION_PREVIOUS: Learner can access the section only after completing the preceding section ORA_SECTION_AFTER_COMPL: Learner can access the section only after completing a specific section identified using the SequenceRuleSectionNumber attribute	You can't use ORA_SECTION_PREVIOUS when no previous section exists. When you use ORA_SECTION_AFTER_COMPL, you need to provide a valid SequenceRuleSectionNumber value. Make sure that you don't define any cyclic dependencies between the sections, such as section A depends on section B and section B depends on section A.
SequenceRuleSectionNumber	Determines the predecessor section that a learner needs to complete before they can access this section when SequenceRuleType is ORA_SECTION_AFTER_COMPL	The number of a section that exists in the parent offering	When SequenceRuleType is ORA_SECTION_AFTER_COMPL, you need to provide a valid SequenceRuleSectionNumber value. Otherwise, the load process ignores any provided value.

How Completion and Sequencing Rules Map to Learning Activity Attributes in CourseV3

These SelfPacedActivityV3 and InstructorLedActivityV3 attributes support activity-level completion and sequencing rules in the CourseV3.dat file.

Attribute	Description	Valid Values	Business Rules
CompletionRuleType	Activity-level completion rule	ORA_REQUIRED or ORA_ OPTIONAL	NA
SequenceRuleType	Determines when a learner can access the Activity.	ORA_ACTIVITY_ANYTIME: Learner can access the section anytime ORA_ACTIVITY_PREVIOUS: Learner can access the section only after completing the preceding section ORA_ACTIVITY_AFTER_COMPL: Learner can access the section only after completing a specific section identified using the SequenceRuleActivityNumber attribute	You can't use ORA_ACTIVITY_ PREVIOUS when no previous section exists. When you use ORA_ ACTIVITY_AFTER_COMPL, you need to provide a valid SequenceRuleActivityNumber value. Make sure that you don't define any cyclic dependencies between the activities, such as activity 1 depends on activity 2 and activity 2 depends on activity 1.
SequenceRuleActivityNumber	It determines the predecessor activity number that learner must complete to be able to access the Activity when SequenceRuleType is ORA_ACTIVITY_AFTER_COMPL.	SequenceRuleActivityNumber should be a valid Activity Number SequenceRuleActivityNumber should exist within the same parent Section	When SequenceRuleType is ORA_ACTIVITY_AFTER_COMPL, you need to provide a valid SequenceRuleActivityNumber value. Otherwise, the load process ignores any provided value.

Guidelines for Effective Start Dates in CourseV3 for Oracle Learning

When loading future-dated course, offering, sections, and activities, use the same effective date. The effective start dates of any sections and activities you're loading for an existing course and offering need to be after those course and offering dates.

Here's an example with the same future effective start date for the course, offering, section, and activity:

```
METADATA|CourseV3|EffectiveStartDate|EffectiveEndDate|CourseNumber|Title|ShortDescription|
Syllabus|PublishStartDate|OwnedByPersonNumber|SourceSystemOwner|SourceSystemId|PublishEndDate|
MinimumExpectedEffort|MaximumExpectedEffort|FLEX:WLF_LI_COURSE|compliance(WLF_LI_COURSE=Global Data
 Elements)|rolerequired(WLF_LI_COURSE=Global Data Elements)|CurrencyCode|MinimumPrice|MaximumPrice
```

```
MERGE|CourseV3|2023/01/24||C_KSP_TEVA_PV|Teva PV Training|Teva PV awareness training 2021 for vendors||
1951/01/01|554872|KNOWME|Course_C_KSP_TEVA_PV||0.1|0.1|Global Data Elements|Yes|Yes|||

METADATA|OfferingV3|EffectiveStartDate|EffectiveEndDate|OfferingStartDate|OfferingEndDate|OfferingNumber|
CoordinatorNumber|Title|OfferingType|PublishStartDate|LanguageCode|CourseNumber|OwnedByPersonNumber|
EnableCapacity|Description|DescriptionText|MinimumCapacity|MaximumCapacity|EnableWaitlist|
PrimaryLocationNumber|PrimaryInstructorNumber|FacilitatorType|SourceSystemOwner|SourceSystemId
MERGE|OfferingV3|2022/01/18||2023/01/24||O_C_KSP_TEVA_PV|554872|Teva PV Training|ORA_SP|1951/01/01|US|
C_KSP_TEVA_PV|554872||DESC_C_KSP_TEVA_PV.txt|Teva PV awareness training 2021 for vendors||||||KNOWME|
COF_O_C_KSP_TEVA_PV_ORA_SP

METADATA|OfferingActivitySectionV3|OfferingSectionId|EffectiveStartDate|EffectiveEndDate|
OfferingSectionNumber|SourceSystemOwner|SourceSystemId|GUID|Title|Description|NumberOfActivitiesToComplete|
CompletionRuleType|OfferingId|SectionPosition|SequenceRuleType|OfferingNumber|DescriptionText|
SequenceRuleSectionNumber
MERGE|OfferingActivitySectionV3||2023/01/24||O_C_KSP_TEVA_PV_SEC|||Default Section|||
ORA_COMPLETION_ACTIVITY||1|ORA_SECTION_ANYTIME|O_C_KSP_TEVA_PV||

METADATA|SelfPacedActivityV3|ActivityId|EffectiveStartDate|EffectiveEndDate|ActivityNumber|
ExpectedEffort|SourceInfo|SourceType|SourceId|SourceSystemOwner|SourceSystemId|GUID|Title|Description|
ContentId|CompletionRuleType|SelfCompleteFlag|OfferingSectionId|ActivityPosition|SequenceRuleType|
OfferingSectionNumber|DescriptionText|ContentNumber|SequenceRuleActivityNumber
MERGE|SelfPacedActivityV3||2023/01/24||O_C_KSP_TEVA_PV_SEC_A01|||||||Activity 3|||ORA_REQUIRED|Y||3||
O_C_KSP_TEVA_PV_SEC|||
```

Here's an example where the effective start dates of the section and activity are after those of the existing course and offering:

```
METADATA|CourseV3|EffectiveStartDate|EffectiveEndDate|CourseNumber|Title|ShortDescription|
Syllabus|PublishStartDate|OwnedByPersonNumber|SourceSystemOwner|SourceSystemId|PublishEndDate|
MinimumExpectedEffort|MaximumExpectedEffort|FLEX:WLF_LI_COURSE|compliance(WLF_LI_COURSE=Global Data
 Elements)|rolerequired(WLF_LI_COURSE=Global Data Elements)|CurrencyCode|MinimumPrice|MaximumPrice
MERGE|CourseV3|1951/01/01||C_KSP_TEVA_PV|Teva PV Training|Teva PV awareness training 2021 for vendors||
1951/01/01|554872|KNOWME|Course_C_KSP_TEVA_PV||0.1|0.1|Global Data Elements|Yes|Yes|||

METADATA|OfferingV3|EffectiveStartDate|EffectiveEndDate|OfferingStartDate|OfferingEndDate|OfferingNumber|
CoordinatorNumber|Title|OfferingType|PublishStartDate|LanguageCode|CourseNumber|OwnedByPersonNumber|
EnableCapacity|Description|DescriptionText|MinimumCapacity|MaximumCapacity|EnableWaitlist|
PrimaryLocationNumber|PrimaryInstructorNumber|FacilitatorType|SourceSystemOwner|SourceSystemId
MERGE|OfferingV3|2022/01/18||2022/11/21||O_C_KSP_TEVA_PV|554872|Teva PV Training|ORA_SP|1951/01/01|US|
C_KSP_TEVA_PV|554872||DESC_C_KSP_TEVA_PV.txt|Teva PV awareness training 2021 for vendors||||||KNOWME|
COF_O_C_KSP_TEVA_PV_ORA_SP

METADATA|OfferingActivitySectionV3|OfferingSectionId|EffectiveStartDate|EffectiveEndDate|
OfferingSectionNumber|SourceSystemOwner|SourceSystemId|GUID|Title|Description|NumberOfActivitiesToComplete|
CompletionRuleType|OfferingId|SectionPosition|SequenceRuleType|OfferingNumber|DescriptionText|
SequenceRuleSectionNumber
MERGE|OfferingActivitySectionV3||2023/01/24||O_C_KSP_TEVA_PV_SEC|||Default Section|||
ORA_COMPLETION_ACTIVITY||1|ORA_SECTION_ANYTIME|O_C_KSP_TEVA_PV||

METADATA|SelfPacedActivityV3|ActivityId|EffectiveStartDate|EffectiveEndDate|ActivityNumber|
ExpectedEffort|SourceInfo|SourceType|SourceId|SourceSystemOwner|SourceSystemId|GUID|Title|Description|
ContentId|CompletionRuleType|SelfCompleteFlag|OfferingSectionId|ActivityPosition|SequenceRuleType|
OfferingSectionNumber|DescriptionText|ContentNumber|SequenceRuleActivityNumber
MERGE|SelfPacedActivityV3||2023/01/24||O_C_KSP_TEVA_PV_SEC_A01|||||||Activity 3|||ORA_REQUIRED|Y||3||
O_C_KSP_TEVA_PV_SEC|||
```

Example of Loading an Entire New Course Hierarchy with a Self-Paced Offering Using the CourseV3.dat File

First you load only the CourseV3 and OfferingV3 objects using the CourseV3.dat file. After that process successfully completes, you load the corresponding sections and activities using the V3 objects and the CourseV3.dat file.

1. Load Only CourseV3 and Self-Paced OfferingV3 Objects

```
METADATA|CourseV3|CourseId|EffectiveStartDate|EffectiveEndDate|CourseNumber|Title|ShortDescription|Syllabus|
PublishStartDate|PublishEndDate|MinimumExpectedEffort|MaximumExpectedEffort|CurrencyCode|MinimumPrice|
MaximumPrice|CoverArtFile|CoverArtFileName|OwnedByPersonId|OwnedByPersonNumber|SourceType|SourceId|
SourceInfo|SourceSystemOwner|SourceSystemId|GUID
MERGE|CourseV3||2021/04/28||COURSE-202104280937-HDL|Title:COURSE 202104280937-HDL|ShortDesc:COURSE
 202104280937-HDL||2021/04/28|2023/03/01|0|1|USD|0|0||||4|FUSION_HCM_LEARNING|2021042809371|HCM Learning
 Source Info for 202104280937|||

METADATA|OfferingV3|OfferingId|EffectiveStartDate|EffectiveEndDate|OfferingNumber|Title|Description|
OfferingType|PublishStartDate|PublishEndDate|OfferingStartDate|OfferingEndDate|PrimaryInstructorId|
FacilitatorType|PrimaryLocationId|CoordinatorNumber|LanguageCode|MinimumCapacity|MaximumCapacity|
CourseId|CourseNumber|OwnedByPersonId|OwnedByPersonNumber|SourceType|SourceId|SourceInfo|
ShowSectionIfSingleSectionExists|SourceSystemOwner|SourceSystemId|GUID
MERGE|OfferingV3||2021/04/28||OFFERING-202104280937-HDL|Title:Offering 202104280937-HDL||ORA_SP|2021/04/28|
2023/03/10||||||4|US|5|30||COURSE-202104280937-HDL||4|FUSION_HCM_LEARNING|2021042809372|HCM Learning Source
 Info for 202104280937|Y|||
```

2. Load V3 Section and Activities Object

```
COMMENT - Include the parent OfferingV3 row even you don't have any updates, otherwise the load process
 fails.
METADATA|OfferingV3|OfferingId|EffectiveStartDate|EffectiveEndDate|OfferingNumber|Title|Description|
OfferingType|PublishStartDate|PublishEndDate|OfferingStartDate|OfferingEndDate|PrimaryInstructorId|
FacilitatorType|PrimaryLocationId|CoordinatorNumber|LanguageCode|MinimumCapacity|MaximumCapacity|
CourseId|CourseNumber|OwnedByPersonId|OwnedByPersonNumber|SourceType|SourceId|SourceInfo|
ShowSectionIfSingleSectionExists|SourceSystemOwner|SourceSystemId|GUID
MERGE|OfferingV3||2021/04/28||OFFERING-202104280937-HDL|Title:Offering 202104280937-HDL||ORA_SP|2021/04/28|
2023/03/10||||||4|US|5|30||COURSE-202104280937-HDL||4|FUSION_HCM_LEARNING|2021042809372|HCM Learning Source
 Info for 202104280937|Y|||

COMMENT - Edit the default section.
METADATA|OfferingActivitySectionV3|OfferingSectionId|EffectiveStartDate|EffectiveEndDate|
OfferingSectionNumber|SectionPosition|OfferingId|OfferingNumber|Title|Description|DescriptionText|
NumberOfActivitiesToComplete|CompletionRuleType|SequenceRuleSectionNumber|SequenceRuleType|
SourceSystemOwner|SourceSystemId|GUID
MERGE|OfferingActivitySectionV3||2021/04/28||OFFERING-202104280937-HDL_SEC|1||OFFERING-202104280937-
HDL||||-1|ORA_COMPLETION_SECTION||ORA_SECTION_ANYTIME|||

COMMENT - Add a self-paced activity to the default section.
METADATA|SelfPacedActivityV3|ActivityId|EffectiveStartDate|EffectiveEndDate|ActivityNumber|ActivityPosition|
Title|Description|ExpectedEffort|SelfCompleteFlag|CompletionRuleType|SequenceRuleActivityNumber|
SequenceRuleType|ContentId|ContentNumber|OfferingSectionId|OfferingSectionNumber|SourceType|SourceId|
SourceInfo|SourceSystemOwner|SourceSystemId|GUID
MERGE|SelfPacedActivityV3||2021/04/28||SP-ACT1-03282122-HDL|1|Title:SP ACT1 202103282122-HDL||8|Y|||
ORA_ACTIVITY_ANYTIME||||OFFERING-202104280937-HDL_SEC||||||

COMMENT - Add a second self-paced activity to the default section.
MERGE|OfferingActivitySectionV3||2021/04/28||OFF-SEC-04012122-HDL|2||OFFERING-202104280937-HDL|Title:OFF-
SEC-04012122-HDL||||ORA_COMPLETION_ACTIVITY||ORA_SECTION_PREVIOUS|||
```

```
MERGE|SelfPacedActivityV3||2021/04/28||SP-ACT2-03282122-HDL|1|Title:SP ACT2 202103282122-HDL||8|N|||
ORA_ACTIVITY_ANYTIME||||OFF-SEC-04012122-HDL||||||

COMMENT - Add an evaluation activity to the offering.
METADATA|EvaluationActivityV3|ActivityId|EffectiveStartDate|EffectiveEndDate|ActivityNumber|Title|
OfferingId|OfferingNumber|QuestionnaireCode|QstnrRequiredForCompletion|SourceSystemOwner|SourceSystemId|GUID
```

Example of Updating a Section Title and the Sequencing Rule Using the CourseV3.dat File

Here's how you can update a section title and change the sequencing rule from previous to a specific section.

```
COMMENT - Include the parent OfferingV3 row even you don't have any updates, otherwise the load process
 fails.
METADATA|OfferingV3|OfferingId|EffectiveStartDate|EffectiveEndDate|OfferingNumber|Title|Description|
OfferingType|PublishStartDate|PublishEndDate|OfferingStartDate|OfferingEndDate|PrimaryInstructorId|
FacilitatorType|PrimaryLocationId|CoordinatorNumber|LanguageCode|MinimumCapacity|MaximumCapacity|
CourseId|CourseNumber|OwnedByPersonId|OwnedByPersonNumber|SourceType|SourceId|SourceInfo|
ShowSectionIfSingleSectionExists|SourceSystemOwner|SourceSystemId|GUID
MERGE|OfferingV3||2021/04/28||OFFERING-202104280937-HDL|Title:Offering 202104280937-HDL||ORA_SP|2021/04/28|
2023/03/10||||||4|US|5|30||COURSE-202104280937-HDL||4|FUSION_HCM_LEARNING|2021042809372|HCM Learning Source
 Info for 202104280937|Y|||

COMMENT - Update the title and change the sequencing rule.
METADATA|OfferingActivitySectionV3|OfferingSectionId|EffectiveStartDate|EffectiveEndDate|
OfferingSectionNumber|SectionPosition|OfferingId|OfferingNumber|Title|Description|DescriptionText|
NumberOfActivitiesToComplete|CompletionRuleType|SequenceRuleSectionNumber|SequenceRuleType|
SourceSystemOwner|SourceSystemId|GUID
MERGE|OfferingActivitySectionV3||2021/04/28||OFF-SEC-04012122-HDL|2||OFFERING-202104280937-HDL|Title:OFF-
SEC-04012122-HDL Updated||||ORA_COMPLETION_ACTIVITY|OFFERING-202104280937-HDL_SEC|ORA_SECTION_AFTER_COMPL|||
```

Example of Updating an Activity Title and Completion Rule Using the CourseV3.dat File

Here's how you can update an activity title and change the completion rule from required to optional.

```
COMMENT - Include the parent OfferingV3 row for the OfferingActivitySectionV3 object even if you don't have
 any updates, otherwise the load process fails.
METADATA|OfferingV3|OfferingId|EffectiveStartDate|EffectiveEndDate|OfferingNumber|Title|Description|
OfferingType|PublishStartDate|PublishEndDate|OfferingStartDate|OfferingEndDate|PrimaryInstructorId|
FacilitatorType|PrimaryLocationId|CoordinatorNumber|LanguageCode|MinimumCapacity|MaximumCapacity|
CourseId|CourseNumber|OwnedByPersonId|OwnedByPersonNumber|SourceType|SourceId|SourceInfo|
ShowSectionIfSingleSectionExists|SourceSystemOwner|SourceSystemId|GUID
MERGE|OfferingV3||2021/04/28||OFFERING-202104280937-HDL|Title:Offering 202104280937-HDL||ORA_SP|2021/04/28|
2023/03/10||||||4|US|5|30||COURSE-202104280937-HDL||4|FUSION_HCM_LEARNING|2021042809372|HCM Learning Source
 Info for 202104280937|Y|||

COMMENT - Include the parent OfferingActivitySectionV3 row for the SelfPacedActivityV3 object even if you
 don't have any updates, otherwise the load process fails.
METADATA|OfferingActivitySectionV3|OfferingSectionId|EffectiveStartDate|EffectiveEndDate|
OfferingSectionNumber|SectionPosition|OfferingId|OfferingNumber|Title|Description|DescriptionText|
NumberOfActivitiesToComplete|CompletionRuleType|SequenceRuleSectionNumber|SequenceRuleType|
SourceSystemOwner|SourceSystemId|GUID
```

```
MERGE|OfferingActivitySectionV3||2021/04/28||OFF-SEC-04012122-HDL|2||OFFERING-202104280937-HDL|Title:OFF-
SEC-04012122-HDL Updated||||ORA_COMPLETION_ACTIVITY|OFFERING-202104280937-HDL_SEC|ORA_SECTION_AFTER_COMPL|||

COMMENT - Update the activity title and completion rule.
METADATA|SelfPacedActivityV3|ActivityId|EffectiveStartDate|EffectiveEndDate|ActivityNumber|ActivityPosition|
Title|Description|ExpectedEffort|SelfCompleteFlag|CompletionRuleType|SequenceRuleActivityNumber|
SequenceRuleType|ContentId|ContentNumber|OfferingSectionId|OfferingSectionNumber|SourceType|SourceId|
SourceInfo|SourceSystemOwner|SourceSystemId|GUID
MERGE|SelfPacedActivityV3||2021/04/28||SP-ACT2-03282122-HDL|1|Title:SP ACT2 202103282122-HDL Updated||8|N|
ORA_REQUIRED||ORA_ACTIVITY_ANYTIME|||OFF-SEC-04012122-HDL||||||
```

Example of Updating Oracle Learning Course Publish End Dates Using HCM Data Loader

To change the publish end dates for courses using HCM Data Loader, you need to get the necessary information for the Course.dat file. Then you create the .dat file and import and load the data.

1. Create and export a report with the appropriate course information.
    a. In Oracle Fusion applications, go to **Tools** > **Reports and Analytics**.
    b. On the Reports and Analytics page, click **Browse Catalog**.
    c. On the Catalog page, on the **New** icon menu, select **Analysis**.
    d. On the Select Subject Area dialog box, search for and select **Workforce Learning - Learning Management Real-Time**.
    e. Expand the Learning Items folder.
    f. Drag and drop **Learning Item Name**in the Selected Columns section.
    g. Click the icon next to Learning Item Name and select **Filter**.
    h. On the New Filter dialog box, click **fx**.
    i. On the Edit Column Formula dialog box, in the Subject Areas pane, expand the Learning Items and Learning Item Type folders.
    j. Select **Code**.
    k. Click the **Add Column** icon. The Column Formula should now look like this: `"Learning Item Type"."Code".`
    l. Click **OK**.
    m. On the New Filter dialog box, in the Value field, search for and select **ORA_COURSE**.
    n. Click **OK**.
    o. On the Untitled page, in the Subject Areas pane, Learning Item Type folder, drag and drop **Learning Item Number**, to the Selected Columns section.
    p. In the Subject Areas pane, expand the **Learning Items** > **Course Specific Details** folder.
    q. Drag and drop **Course Effective Start Date** to the Selected Columns area.
    r. In the Subject Areas pane, expand the **Learning Items** > **Learning Item Owner** > **Worker (Owner)** folder.
    s. Drag and drop **Person Number** to the Selected Columns area.
    t. Click the Results tab.
    u. On the Export this analysis icon menu, select **Data** > **Tab Delimited**.
    v. Save the file to your device.
    w. On the Confirmation dialog box, click **OK**.

2. Create the Course.dat file.
    a. Open a blank Microsoft Excel file.
    b. Copy this content and paste it into cell 1A.

```
METADATA|Course|CourseId|EffectiveStartDate|EffectiveEndDate|CourseNumber|Title|ShortDescription|
Syllabus|PublishStartDate|PublishEndDate|MinimumExpectedEffort|MaximumExpectedEffort|CurrencyCode|
MinimumPrice|MaximumPrice|CoverArtFile|CoverArtFileName|TrailerLiId|TrailerLiNumber|OwnedByPersonId|
OwnedByPersonNumber|SourceType|SourceId|SourceInfo|SourceSystemOwner|SourceSystemId|GUID
MERGE|Course|||||||||||||||||||||||||
```

    c. Select cell 1A.
    d. On the Data tool strip, click **Text to Columns**.
    e. On the Convert Text to Columns Wizard - Step 1 of 3 dialog box, accept the defaults by clicking **Next**.
    f. On the Convert Text to Columns Wizard - Step 2 of 3 dialog box, in the Delimiters group, select **Other**.
    g. In the corresponding field enter only the pipe symbol (|).
    h. Click **Next**.
    i. On the Convert Text to Columns Wizard - Step 3 of 3 dialog box, click **Finish**.
    j. Add a row for each course with a publish end date to update.

        - The METADATA value needs to be MERGE.
        - Dates need to follow the YYYY/MM/DD format.
        - The EffectiveStartDate cell is for the course effective start date.
        - You can find the course publish start dates in using the Courses task on the **My Client Groups** > **Learning** page. They're in the Start Date column.
        - For OwnedbyPersonNumber, use the person number from your exported analysis.

    Here's how the column headers should look and a sample row of a course to update:

```
METADATA|Course|CourseId|EffectiveStartDate|EffectiveEndDate|CourseNumber|Title|ShortDescription|
Syllabus|PublishStartDate|PublishEndDate|MinimumExpectedEffort|MaximumExpectedEffort|CurrencyCode|
MinimumPrice|MaximumPrice|CoverArtFile|CoverArtFileName|TrailerLiId|TrailerLiNumber|OwnedByPersonId|
OwnedByPersonNumber|SourceType|SourceId|SourceInfo|SourceSystemOwner|SourceSystemId|GUID
MERGE|Course||2019/06/14||OLCxxxxxx|Example Course|||2019/06/14|2019/07/22|||||||||||xxxxxx||||||
```

    k. After you add all of the courses to update, in the Microsoft Excel file, click **File** > **Save as**.
    l. On the Save As dialog box, in the File name field, enter **Course**.
    m. In the Save as type field, select **Text (Tab delimited)(*.txt)**.
    n. On the Microsoft Excel dialog box asking if you want to keep the format, click **Yes**.
    o. Close the Microsoft Excel file.
    p. Go to the location where you saved the file.
    q. Change the file name from Course.txt to **Course.dat**.
    r. Right-click the Course.dat file and select **Open with** > **Notepad++**. You can download and install the app from your browser search if you don't already have it.
    s. In Notepad++, click **Search** > **Replace**.
    t. In the Find what field, enter **\t**
    u. In the Replace field, enter **|**
    v. In the Search Mode group, select **Extended (\n, \r, \t, \0, \x...)**.
    w. Click **Replace All**. This makes sure that the pipe (|) is the delimiter, which HCM Data Loader requires.
    x. Save your changes and close the file.
    y. Zip the file with the name Course.zip

3. Import the course data using HCM Data Loader.
    a. In Oracle Fusion apps, click **My Client Groups Data Exchange**.
    b. On the Data Exchange page, search for and click the **Import and Load Data** task.
    c. On the Import and Load Data page, click **Import File**.
    d. On the Import File dialog box, browse for and select your **Course.zip** file.
    e. Review the information.
    f. Click **Submit Now**.
    g. Refresh the Import and Load Data page to make sure the file loaded without error.

Guidelines for Loading a Course to Oracle Learning Using Course

If you haven't turned on activity sequencing, you can load courses, offerings, activities, and classroom and instructor reservations to the Oracle Learning catalog using the Course object. If you have it turned on, use the CourseV3 object.

To verify that the course successfully loaded, use the Courses task on the **My Client Groups** > **Learning** page. To verify that an offering, activities, and classroom and instructor reservations successfully loaded, use the Offerings task.

Here's the hierarchy and processing order for the relevant learning items:

1. ClassroomResource (processing order: 106030)
2. ClassroomResourceTranslation (processing order 106031)
3. Course (processing order: 106060)
    a. Offering
        i. InstructorLedActivity
            a. ClassroomReservation
            b. InstructorReservation
            c. AdhocResource
        ii. SelfPacedActivity
        iii. Offering Default Access
4. Course Default Access CourseTranslation (processing order: 106061)
5. OfferingTranslation (processing order: 106062)
6. InstructorLedActivityTranslation (processing order: 106064)
7. SelfPacedActivityTranslation (processing order: 106065)
8. CourseOfferingPricingDefaults (processing order: 106070)
    a. CourseOfferingPricingComponent
9. OfferingCustomPricing (processing order: 106075)
    a. OfferingCustomPricingComponent

Rich Text Syllabus and Cover Art

A course file can have a rich text syllabus description and cover art.

- To load rich text, you need to reference a rich text file in the Syllabus attribute, such as an .html file. To load the rich text files, place them all in a folder called ClobFiles and zip the folder. Then include that zip inside the Course.zip file.
- To load cover art, you need to reference an image file in the CoverArtFile and CoverArtFileName attributes, such as a .jpg file. To load the image files, place them in a folder called BlobFiles and zip the folder. Then include that .zip file inside the Course.zip file.

Validations

The load process does these validations:

- On create, the CourseNumber value doesn't exist and the Title and OwnedByPersonID values aren't null
- The Person object exists for the provided OwnedByPersonId value
- The Trailer Video record for the provided TrailerLiId value exists in Oracle Learning
- Checks if a CoverArtFile is provided and if so, that the CoverArtFile extension is one of these valid file formats: .jpeg, .jpg, .bmp, .tiff, .png, or .gif, and can be read
- The EffectiveStartDate value isn't empty or null
- The PublishStartDate value is before the PublishEndDate value
- The MinimumExpectedEffort value is either 0 or a positive number
- The MaximumExpectedEffort, MinimumPrice, and MaximumPrice values are a positive number
- The MinimumExpectedEffort value is less than the MaximumExpectedEffort value
- The MinimumPrice value is less than the MaximumPrice value
- It can retrieve the access properties for the learning item

Example of Loading an Oracle Learning Course Using Course

Here's how you can load a Course object using user keys.

```
METADATA|Course|EffectiveStartDate|CourseNumber|Title|ShortDescription|PublishStartDate|PublishEndDate|
MinimumExpectedEffort|MaximumExpectedEffort|CurrencyCode|MinimumPrice|MaximumPrice|TrailerLiNumber|
OwnedByPersonNumber|SourceType|SourceId|SourceInfo
MERGE|Course|2018/02/04|COURSE-201802041711-HDL|Title:COURSE 201802041711|ShortDesc:COURSE201802041711|
2018/02/04|2019/02/03|0|1|USD|0|0|OLC103778|8153757|FUSION_HCM_LEARNING|201802041711|HCM Learning Source
 Info for 201802041711 HDL Update
```

Guidelines for Loading Offerings to Oracle Learning

You can load self-paced, instructor-led (ILT) or blended offerings using the Offering component of the Course object. The course that an offering links to must exist in Oracle Learning before you load the offering.

To verify that the course successfully loaded, use the Offerings task on the **My Client Groups** > **Learning** page.

General Validations

The load process does these validations on create:

- The PersonId value isn't null and the Person and InstructorResource objects already exist for the provided PersonId value
- The InstructorResourceNumber value exists in Oracle Learning
- The OwnedByPersonId value isn't null and the Person object already exists for the provided OwnedByPersonId value
- The OfferingNumber value doesn't already exist in Oracle Learning
- The Title, OfferingType, and CourseId values aren't null

It also does these validations:

- The Person object exists for the provided PersonID or OwnedByPersonId values
- The course details for the provide CourseLearningItemId value exist in Oracle Learning
- The Questionnaire record for the provided QuestionnaireCode value exists in Oracle Learning
- A default questionnaire for the provided OfferingId value doesn't exist
- The PublishStartDate value is before the PublishEndDate value
- The offering PublishStartDate value is after the course PublishStartDate value
- The MinimumCapacity value is 0 or a positive whole number
- The MaximumCapacity value is a positive whole number
- The MinimumCapacity value is less than the MaximumCapacity value
- The EnableCapacity and EnableWaitList values are either 'Y' and 'N'
- The AccessPermissionId and TrainingSupplierId values exist in Oracle Learning
- The FacilitatorType value is ORA_TRNG_VENDOR or ORA_INSTRUCTOR
- The values for these attributes aren't empty or null:
    - EffectiveStartDate
    - OfferingStartDate
    - OfferingEndDate
    - MaximumCapacity when the EnableCapacity value is 'Y'
    - PrimaryLocation ILT and blended offerings
        - For virtual classrooms, set the PrimaryLocationId value -1
        - For not defined classrooms, set the PrimaryLocationId value -2

- Coordinator
- FacilitatorType when you provide either a PrimaryInstructorId or TrainingSupplierId value
- TrainingSupplierId when the FacilitatorType value is ORA_TRNG_VENDOR
- PrimaryInstructorId when FacilitatorType is ORA_INSTRUCTOR

- An offering has only one evaluation activity otherwise the process ends in error
  - To use the default evaluation, for QuestionnaireCode, enter -1. To use a specific evaluation, enter its questionnaire code.

Example of Loading an Offering to Oracle Learning Using Offering

Here's how you can load Offering components using user keys.

```
METADATA|Offering|EffectiveStartDate|OfferingNumber|Title|OfferingType|PublishStartDate|PublishEndDate|
LanguageCode|MinimumCapacity|MaximumCapacity|QuestionnaireCode|QstnrRequiredForCompletion|CourseNumber|
OwnedByPersonNumber|SourceType|SourceId|SourceInfo
MERGE|Offering|2018/02/04|OFFERING-201802041712-HDL|Title:Offering 201802041712|ORA_BLENDED|2018/02/04|
2019/02/03|US|5|30|-1|N|COURSE-201802041711-HDL|8153757|FUSION_HCM_LEARNING|201802041712|HCM Learning Source
 Info for 201802041712
```

Guidelines for Loading the Custom Pricing of an Oracle Learning Offering

You can load custom pricing for a course offering using the OfferingCustomPricing object and OfferingCustomPricingComponent component. The offering that the custom pricing is for needs to exist in Oracle Learning before you load the pricing.

To verify that the self-paced activity successfully loaded, use the Offerings task on the **My Client Groups** > **Learning** page.

The PricingComponentType attribute is part of the OfferingCustomPricingComponent component. Valid values are defined in the ORA_WLF_PRICING_TYPE lookup type, which includes these delivered lookups:

Value	Meaning
ORA_LIST_PRICE	List price
ORA_LIST_PRICE_ADJ	List price adjustment

Examples of Loading the Custom Pricing of an Oracle Learning Offering

Here's how you can load custom pricing for offerings to Oracle Learning using user keys.

1. Create Offering Custom Pricing Objects

This example OfferingCustomPricing.dat file identifies the OfferingCustomPricing object by its user key.

```
METADATA|OfferingCustomPricing|EffectiveStartDate|PricingRuleNumber|CurrencyCode|OfferingNumber
MERGE|OfferingCustomPricing|2018/02/04|PR-201802041723-HDL|USD|OFFERING-201802041712-HDL
```

2. Create Offering Custom Pricing Components

This example OfferingCustomPricing.dat file loads two OfferingCustomPricingComponent components using their user keys.

```
METADATA|OfferingCustomPricingComponent|EffectiveStartDate|PricingComponentNumber|PricingComponentType|
Price|IncludeInSelfServicePricing|PricingRuleId|PricingRuleNumber
MERGE|OfferingCustomPricingComponent|2018/02/04|PC-201802041724-HDL|ORA_LIST_PRICE|100|N|PR-201802041723-HDL
MERGE|OfferingCustomPricingComponent|2018/02/04|PC-201802041725-HDL|ORA_LIST_PRICE_ADJ|200|N|
PR-201802041723-HDL
```

38 Loading Specializations to the Learning Catalog

Guidelines for Loading Specializations to Oracle Learning Using SpecializationV3

You can load specializations to the Oracle Learning catalog. If you have activity sequencing turned on, use the SpecializationV3 object. If it's turned off, use Specialization.

> **Tip:** Unused Attributes can get deprecated, so always use the most recent .dat file.

To verify that the specialization successfully loaded, use the Specialization task on the **My Client Groups** > **Learning** page.

Here's the hierarchy and processing order for these learning objects:

1. SpecializationV3
    a. SpecializationDefaultAccessV3
    b. SpecializationSectionV3
        i. SpecializationSectionActivityV3
2. SpecializationV3Translation
3. SpecializationSectionV3Translation

When you create the specialization hierarchy, you need to create the sections and activities in a separate DAT file from the specializations. Here's how you create the complete specialization hierarchy:

1. Load the SpecializationV3 object using a single SpecializationV3.dat file.

    Make sure that the SpecializationNumber doesn't exceed 26 characters. Otherwise the autogenerated SectionNumber for the default section will exceed 30 characters and result in an HCM Data Loader error.
2. Load the SpecializationSectionV3 and SpecializationSectionActivityV3 components together in a separate SpecializationV3.dat file.
    - Make sure that the .dat file also includes the parent SpecializationV3 reference, even if you aren't updating the parent. Otherwise the load process fails with an error.
    - For all activities, include the parent SpecializationSectionV3 row in the dat file.

Default Section

The load process automatically does this work for the default section of an offering:

- Sets SectionTitle to Default Section.
- Automatically generates the SectionNumber value using the parent SpecializationNumber and the _SEC suffix. For example, if the offering number is SPECIAL-202104280937-HDL, then the generated section number is OFFERING-202104280937-HDL_SEC.

- Sets Position to 1. If you add more sections to the specialization, set the Position values in the appropriate sequence, starting at 2.
- Sets ShowSectionIfSingleSectionExists to N so that the section isn't visible to learners. If you add more sections to the specialization, the load process ignores any value for this attribute. This attribute is equivalent to the Enable as a visible section to learners option on the Activities tab of the specialization details page.

SpecializationV3 DAT File Structure with Attributes

```
METADATA|SpecializationV3|SpecializationId|EffectiveStartDate|EffectiveEndDate|SpecializationNumber|Title|
ShortDescription|Description|PublishStartDate|PublishEndDate|CoverArtFile|CoverArtFileName|SourceType|
SourceId|SourceInfo|OwnedByPersonId|OwnedByPersonNumber|MinimumExpectedEffort|MaximumExpectedEffort|
SourceSystemOwner|SourceSystemId|GUID|DescriptionText|ShowSectionIfSingleSectionExists

METADATA|SpecializationDefaultAccessV3|DefaultAccessId|DefaultAccessNumber|EffectiveStartDate|
EffectiveEndDate|FollowCommunity|SelfViewMode|SelfInitialStatus|SelfEnrollForm|SelfActivateApprove|
SelfEnrollQuestionnaire|SelfQuestionnaireCode|SelfPrereqType|SelfWithdrawPrereqDays|MgrInitialStatus|
MgrActivateApprove|MgrWithdrawPrereqDays|MgrMarkComplete|SourceSystemId|SourceSystemOwner|GUID|
SpecializationId|SpecializationNumber

METADATA|SpecializationSectionV3|SpecializationSectionId|SpecializationSectionNumber|EffectiveStartDate|
EffectiveEndDate|Title|Description|DescriptionText|CompletionRuleType|NumberOfActivitiesToComplete|
SequenceRuleType|SequenceRuleSectionNumber|InitialAssignmentStatusOfActivities|SectionPosition|
SpecializationId|SpecializationNumber|SourceSystemOwner|SourceSystemId|GUID

METADATA|SpecializationSectionActivityV3|ActivityId|EffectiveStartDate|EffectiveEndDate|ActivityNumber|
SpecializationSectionId|SpecializationSectionNumber|CourseLearningItemId|CourseLearningItemNumber|
SequenceRuleType|SequenceRuleActivityNumber|ActivityPosition|ActivityCompletionType|SourceSystemId|
SourceSystemOwner|GUID
```

How Completion and Sequencing Rules Map to Learning Section Attributes in SpecializationV3

These SpecializationV3 attributes support section-level completion and sequencing rules in the SpecializationV3.dat file.

Attribute	Description	Valid Values	Business Rules
CompletionRuleType	Determines whether the completion rule is defined at the Section or Activity level	ORA_COMPLETION_ACTIVITY or ORA_COMPLETION_SECTION	When you use ORA_COMPLETION_SECTION, you need to provide a valid NumberOfActivitiesToComplete value.
NumberOfActivitiesToComplete	Number of activities a learner needs to complete, to complete the section when CompletionRuleType is Section	When all section activities are required: -1 Otherwise a numeric value between 0 and n where n is the total number of section activities.	The specified value can't exceed the total number of activities. When CompletionRuleType is ORA_COMPLETION_ACTIVITY, the load process ignores any value that you provide.

Attribute	Description	Valid Values	Business Rules
SequenceRuleType	Determines when a learner can access the section	ORA_SECTION_ANYTIME: Learner can access the section anytime ORA_SECTION_PREVIOUS: Learner can access the section only after completing the preceding section ORA_SECTION_AFTER_COMPL: Learner can access the section only after completing a specific section identified using the SequenceRuleSectionNumber attribute	You can't use ORA_SECTION_PREVIOUS when no previous section exists. When you use ORA_SECTION_AFTER_COMPL, you need to provide a valid SequenceRuleSectionNumber value. Make sure that you don't define any cyclic dependencies between the sections, such as section A depends on section B and section B depends on section A.
SequenceRuleSectionNumber	Determines the predecessor section that a learner needs to complete before they can access this section when SequenceRuleType is ORA_SECTION_AFTER_COMPL	The number of a section that exists in the parent offering	When SequenceRuleType is ORA_SECTION_AFTER_COMPL, you need to provide a valid SequenceRuleSectionNumber value. Otherwise, the load process ignores any provided value.

How Completion and Sequencing Rules Map to Learning Activity Attributes in SpecializationV3

These SelfPacedActivityV3 and InstructorLedActivityV3 attributes support activity-level completion and sequencing rules in the SpecializationV3.dat file.

Attribute	Description	Valid Values	Business Rules
CompletionRuleType	Activity-level completion rule	ORA_REQUIRED or ORA_OPTIONAL	NA
SequenceRuleType	Determines when a learner can access the Activity.	ORA_ACTIVITY_ANYTIME: Learner can access the section anytime ORA_ACTIVITY_PREVIOUS: Learner can access the section only after completing the preceding section ORA_ACTIVITY_AFTER_COMPL: Learner can access the section only after completing a specific section identified using the SequenceRuleActivityNumber attribute	You can't use ORA_ACTIVITY_PREVIOUS when no previous section exists. When you use ORA_ACTIVITY_AFTER_COMPL, you need to provide a valid SequenceRuleActivityNumber value. Make sure that you don't define any cyclic dependencies between the activities, such as activity 1 depends on activity 2 and activity 2 depends on activity 1.

Attribute	Description	Valid Values	Business Rules
SequenceRuleActivityNumber	It determines the predecessor activity number that learner must complete to be able to access the Activity when SequenceRuleType is ORA_ACTIVITY_AFTER_COMPL.	SequenceRuleActivityNumber should be a valid Activity Number SequenceRuleActivityNumber should exist within the same parent Section	When SequenceRuleType is ORA_ACTIVITY_AFTER_COMPL, you need to provide a valid SequenceRuleActivityNumber value. Otherwise, the load

Example for Loading an Entire New Specialization Hierarchy with Sections and Activities Using the SpecializationV3.dat File

First you load only the SpecializationV3 object using the SpecializationV3.dat file. After that process successfully completes, you load the corresponding sections and activities using the V3 objects and the SpecializationV3.dat file.

1. Load Only the SpecializationV3 Object

```
METADATA|SpecializationV3|SpecializationId|EffectiveStartDate|EffectiveEndDate|SpecializationNumber|Title|
ShortDescription|Description|PublishStartDate|PublishEndDate|CoverArtFile|CoverArtFileName|SourceType|
SourceId|SourceInfo|OwnedByPersonId|OwnedByPersonNumber|MinimumExpectedEffort|MaximumExpectedEffort|
SourceSystemOwner|SourceSystemId|GUID|DescriptionText|ShowSectionIfSingleSectionExists
MERGE|SpecializationV3||2021/08/02||HDLSPEC1_451_0208|SpecializationV3 HDL 0208|Test Short Description||
2021/08/02||||||||4|15|30|||||Y
```

2. Load V3 Section and Activities Objects

```
COMMENT - Include the parent SpecializationV3 row even you don't have any updates, otherwise the load
 process fails.
METADATA|SpecializationV3|SpecializationId|EffectiveStartDate|EffectiveEndDate|SpecializationNumber|Title|
ShortDescription|Description|PublishStartDate|PublishEndDate|CoverArtFile|CoverArtFileName|SourceType|
SourceId|SourceInfo|OwnedByPersonId|OwnedByPersonNumber|MinimumExpectedEffort|MaximumExpectedEffort|
SourceSystemOwner|SourceSystemId|GUID|DescriptionText|ShowSectionIfSingleSectionExists
MERGE|SpecializationV3||2021/08/02||HDLSPEC1_451_0208|SpecializationV3 HDL 0208|Test Short Description||
2021/08/02||||||||4|15|30|||||Y

COMMENT - Edit the default section.
METADATA|SpecializationV3|SpecializationId|EffectiveStartDate|EffectiveEndDate|SpecializationNumber|Title|
ShortDescription|Description|PublishStartDate|PublishEndDate|CoverArtFile|CoverArtFileName|SourceType|
SourceId|SourceInfo|OwnedByPersonId|OwnedByPersonNumber|MinimumExpectedEffort|MaximumExpectedEffort|
SourceSystemOwner|SourceSystemId|GUID|DescriptionText|ShowSectionIfSingleSectionExists
MERGE|SpecializationV3||2021/08/02||HDLSPEC1_451_0208|SpecializationV3 HDL 0208|Test Short Description||
2021/08/02||||||||4|15|30|||||Y

COMMENT - Add another section.
METADATA|SpecializationSectionV3|SpecializationSectionId|EffectiveStartDate|EffectiveEndDate|
SpecializationSectionNumber|Title|Description|DescriptionText|CompletionRuleType|
NumberOfActivitiesToComplete|SequenceRuleType|SequenceRuleSectionNumber|InitialAssignmentStatusOfActivities|
SectionPosition|SpecializationId|SpecializationNumber
MERGE|SpecializationSectionV3||2021/08/02||HDLSPEC1_451_0208_SEC|Default Section||Update default
 section||||||1||HDLSPEC1_451_0208
MERGE|SpecializationSectionV3||2021/08/02||HDLSPEC1_451_SEC2|Section 2||Description::Section2|
ORA_COMPLETION_SECTION|-1|ORA_SECTION_PREVIOUS||ORA_ASSN_REC_ACTIVE|2||HDLSPEC1_451_0208
```

```
COMMENT - Add activities to the second section.
METADATA|SpecializationSectionActivityV3|ActivityId|EffectiveStartDate|EffectiveEndDate|ActivityNumber|
SpecializationSectionId|SpecializationSectionNumber|CourseLearningItemId|CourseLearningItemNumber|
SequenceRuleType|SequenceRuleActivityNumber|ActivityPosition|ActivityCompletionType
MERGE|SpecializationSectionActivityV3||2021/08/02||HDLSPEC1_451_S1_A1||HDLSPEC1_451_0208_SEC||
COURSE-202104280937-HDL|||1|ORA_REQUIRED
```

Guidelines for Loading Specializations to Oracle Learning Using Specializations

You can load specializations to the Oracle Learning catalog. If you have activity sequencing turned on, use the SpecializationV3 object. If it's turned off, use Specialization.

To verify that the specialization successfully loaded, use the Specialization task on the **My Client Groups** > **Learning** page.

Here's the hierarchy and processing order for these learning objects:

1. Specialization (processing order: 106080)
    a. SpecializationDefaultAccess
    b. SpecializationSection
        i. SpecializationSectionActivity
2. SpecializationTranslation (processing order: 106081)
3. SpecializationSectionTranslation (processing order: 106082)

Specialization objects that you create using HCM Data Loader have the default access values defined on the Manage Default Access page of the specialization.

The default value for the NumberOfActivitiesToComplete attribute of the SpecializationSection component is -1. This means that all activities in the section must be completed.

Example of Loading a Specialization to Oracle Learning Using Specialization

Here's how you create a Specialization object, two SpecializationSection components, and two SpecializationSectionActivity components. The Specialization object is identified using source keys.

```
COMMENT Specialization object
METADATA|Specialization|EffectiveStartDate|SpecializationNumber|Title|ShortDescription|SourceType|SourceId|
SourceInfo|OwnedByPersonNumber|SourceSystemOwner|SourceSystemId
MERGE|Specialization|2018/03/05|DSS-2018043009000-HDL|Title: DSS HDL Specialization 2018043009000|SD: DSS
 HDL Specialization 2018043009000 Short Description|FUSION_HCM_LEARNING|2018043009000|HCM Learning Source
 Information for 2018043009000|8153756|VISION|2018043009000

COMMENT SpecializationSection components
METADATA|SpecializationSection|EffectiveStartDate|SpecializationSectionNumber|Title|SectionPosition|
NumberOfActivitiesToComplete|InitialAssignmentStatusOfActivities|SpecializationNumber
```

```
MERGE|SpecializationSection|2018/03/05|Sec-2018043009000-HDL|Title: DSS HDL Section# 1|1|0|
ORA_ASSN_REC_ACTIVE|DSS-2018043009000-HDL
MERGE|SpecializationSection|2018/03/05|Sec-2018043009001-HDL|Title: DSS HDL Section# 2|2|0|
ORA_ASSN_REC_ACTIVE|DSS-2018043009000-HDL

COMMENT SpecializationSectionActivity components
METADATA|SpecializationSectionActivity|EffectiveStartDate|ActivityNumber|ActivityPosition|
SpecializationSectionNumber|LearningItemNumber
MERGE|SpecializationSectionActivity|2018/03/05|SecAct-2018043009000-HDL|1|Sec-2018043009000-HDL|OLC900002
MERGE|SpecializationSectionActivity|2018/03/05|SecAct-2018043009001-HDL|1|Sec-2018043009001-HDL|
COURSE-2018040910351-HDL
```

39 Loading Default Access for Learning Courses, Offerings, and Specializations

Guidelines for Loading the Default Access of an Oracle Learning Course

You can load the default access properties of a course using the Course Default Access object. The access properties apply to all learners accessing courses who aren't part of an access group and don't have a learning assignment.

To verify that the course successfully loaded, use the Courses task on the **My Client Groups** > **Learning** page. Open the course details page and click **Manage Default Access**.

Guidelines for Loading the Default Access of an Oracle Learning Offering

You can load the default access properties of an offering using the Offering Default Access object. The access properties apply to all learners accessing offerings who aren't part of an access group and don't have a learning assignment.

To verify that the course successfully loaded, use the Offerings task on the **My Client Groups** > **Learning** page. Open the offering details page and click **Manage Default Access**.

Guidelines for Loading the Default Access of an Oracle Learning Specialization

You can load the default access properties of a specialization using the Specialization Default Access object. The access properties apply to all learners accessing specializations who aren't part of an access group and don't have a learning assignment.

To verify that the course successfully loaded, use the Specializations task on the **My Client Groups** > **Learning** page. Open the specialization details page and click **Manage Default Access**.

40 Loading Offering Activities to the Learning Catalog

Guidelines for Loading an Instructor-Led Training (ILT) Activity for an Oracle Learning Offering

You can load an instructor-led training activity for an ILT or blended offering. If you have activity sequencing turned on, use the InstructorLedActivityV3 object. If it's turned off, use the InstructorLedActivity component.

The offering that the activity is part of needs to exist in Oracle Learning before you load the activity.

The InstructorLedActivityV3 object and InstructorLedActivity component have child components that identify resources required for the activity.

- You can identify a single classroom resource on the InstructorLedActivity component itself. To identify additional classroom resources, use the ClassRoomReservation component. The ClassRoomResource object that the ClassRoomReservation component references must exist.
- To add instructors, use the InstructorReservation component. The InstructorResource object that the InstructorReservation component references must exist

To verify that the ILT activity successfully loaded, use the Offerings task on the **My Client Groups** > **Learning** page. On the offering details page, click the Activities tab.

Validations

The load process does these validations:

- Either the VirtualClassroomURL attribute (indicates that the activity is online) or the ClassroomResourceNumber attribute (indicates that the activity is onsite) has a valid value. If you provide values for both attributes, you're indicating that the activity is onsite.
- The ActivityStartTime and ActivityEndTime values are in the 24-hour format, such as 10:00:00 or 19:00:00.
- The Timezone value is in the TZ database names format, such as America/Los_Angeles.

Example of Loading an ILT Activity for an Oracle Learning Offering

Here's how you can load InstructorLedActivity objects using user keys. This sample also shows you how to load child objects that support the activity, such as ClassroomReservation and InstructorReservation.

```
METADATA|InstructorLedActivity|EffectiveStartDate|ActivityNumber|Title|ActivityDate|ActivityStartTime|
ActivityEndTime|TimeZone|ExpectedEffort|SelfCompleteFlag|OfferingNumber|VirtualClassroomUrl|
ClassroomResourceNumber|SourceType|SourceId|SourceInfo
```

```
MERGE|InstructorLedActivity|2018/02/04|ILT-ACT-201802041713-HDL|Title:ILT ACT 201802041713|2018/02/04|
10:00:00|19:00:00|America/Los_Angeles|30|Y|OFFERING-201802041712-HDL|http://www.example.com|
RSC-201802041710-HDL|FUSION_HCM_LEARNING|201802041713|HCM Learning Source Info for 201802041713

COMMENT Load two Ad Hoc Resource components
METADATA|AdhocResource|AdhocResourceNumber|Name|Description|Quantity|ActivityNumber
MERGE|AdhocResource|ADHOC-201802041714-HDL|SS-ADHOC-RESOURCE-1|SS-ADHOC-RESOURCE-1 DESC|30|ILT-
ACT-201802041713-HDL
MERGE|AdhocResource|ADHOC-201802041715-HDL|SS-ADHOC-RESOURCE-2|SS-ADHOC-RESOURCE-2 DESC|30|ILT-
ACT-201802041713-HDL

COMMENT Load one Classroom Reservation component
METADATA|ClassroomReservation|ClassroomReservationNumber|ClassroomResourceNumber|ActivityNumber
MERGE|ClassroomReservation|CRSV-201802041716-HDL|OLC136311|ILT-ACT-201802041713-HDL

COMMENT Load two Instructor Reservation components
METADATA|InstructorReservation|InstructorReservationNumber|InstructorResourceNumber|ActivityNumber
MERGE|InstructorReservation|INSTR-201802041717-HDL|8153762|ILT-ACT-201802041713-HDL
MERGE|InstructorReservation|INSTR-201802041718-HDL|8153771|ILT-ACT-201802041713-HDL
```

Guidelines for Loading Ad Hoc Resources for ILT Offering Activities in Oracle Learning

You can load unplanned teaching resources for on-site instructor-led training (ILT) activities, such as flip charts, projectors, and pens. If you have activity sequencing turned on, use the AdhocResourceV3 object. If it's turned off, use AdhocResource.

The ILT activity that the resources are for needs to exist in Oracle Learning before you load the ad hoc resources. To verify that the ad hoc resources successfully loaded, use the Offerings task on the **My Client Groups** > **Learning** page. On the offering details page, click the Activities tab. Edit the applicable activity and review the Other Resources section.

Guidelines for Loading Instructor Reservations for an ILT Offering Activity in Oracle Learning

You can load instructor reservations for an instructor-led training (ILT) activities. If you have activity sequencing turned on, use the InstructorReservationV3 object. If it's turned off, use InstructorReservation.

The ILT activity that the instructor reservations are for needs to exist in Oracle Learning before you load the reservations. To verify that the improvised resources successfully loaded, use the Offerings task on the **My Client Groups** > **Learning** page. On the offering details page, click the Activities tab. Edit the applicable activity and review the Instructors section.

Guidelines for Loading a Self-Paced Offering Activity for an Oracle Learning Offering

You can load a self-paced activity for blended and self-paced offerings. If you have activity sequencing turned on, use the SelfPacedActivityV3 object. If it's turned off, use the SelfPacedActivity component.

The offering that the activity is part of needs to exist in Oracle Learning before you load the activity.

Supplying a value for the ContentNumber attribute indicates a self-paced, online activity. Otherwise, the activity is self-paced, offline.

To verify that the self-paced activity successfully loaded, use the Offerings task on the **My Client Groups** > **Learning** page. On the offering details page, click the Activities tab.

Example of Loading a Self-Paced Activity for an Oracle Learning Offering

Here's how you can load SelfPacedActivity components using user keys.

```
METADATA|SelfPacedActivity|EffectiveStartDate|ActivityNumber|ActivityPosition|Title|ExpectedEffort|
SelfCompleteFlag|ContentNumber|OfferingNumber|SourceType|SourceId|SourceInfo
MERGE|SelfPacedActivity|2018/02/04|SP-ACT-201802041719-HDL|1|Title:SP ACT 201802041719|8|N|OLC103580|
OFFERING-201802041712-HDL|FUSION_HCM_LEARNING|201802041719|HCM Learning Source Info for 201802041719
```

41 Loading Learning Assignments and Recommendations

Guidelines for Loading an Assignment or Recommendation to Oracle Learning

You can assign specific learning to an individual learner and update existing assignment statuses using the Learning Record object. Also use it to recommend specific learning and update existing recommendation statuses.

You can create learning assignments for all valid learning item types:

- Legacy (ORA_LEGACY)
- Course (ORA_COURSE)
- Offering (ORA_CLASS)
- Specialization (ORA_SPECIALIZATION)
- elearning (ORA_ELEARNING)
- Noncatalog (ORA_NONCATALOG)

To verify that the learning assignment successfully loaded and assignment statuses successfully changed, use the Learning Assignments task on the **My Client Groups** > **Learning** page.

Toolkit for Updating LearningRecord

Because the LearningRecord object is the most loaded object for Oracle Learning, we created two Oracle Business Intelligence (BI) reports.

- One report uses a learning item number as an input to get the full set of assignment records in the correct format.
- The other report uses the assignment record number as an input to get the assignment record in the correct format. You can provide more than one assignment record number–separated by commas--in the parameter.

Here's what you can do with these reports:

1. Run a report.
2. Change the values in the generated file.
3. Zip the file.
4. Load the changes using the LearningRecord object and HCM Data Loader.

For information about accessing the BI reports and more detailed instructions for using them, see Oracle Fusion Learning Cloud: How to Delete Completed Learning Assignment with HCM Data Loader (HCM Data Loader) (document ID *2575333.1*) on My Oracle Support.

General Validations Done When Loading Oracle Learning Assignments and Recommendations

The HCM Data Loader process for the LearningRecord object does these general validations:

- The values for these attributes aren't null:
    - AssignmentNumber
    - LearningRecordNumber
    - EffectiveStartDate
    - LearningItemType
    - LearningItemNumber
    - AssignmentType
    - AssignmentSubType
    - AssignedByPersonNumber
    - AssignmentAttributionType
    - AssignmentAttributionNumber
    - AssignmentAttributionCode
    - LearnerNumber
        - LearningRecordStatus
        - LearningRecordStartDate
- The LearningRecordReasonCode and LearningRecordComments values aren't null when the learning assignment status is Withdrawn or Deleted. They also can't be null when the status is Completed and AssignmentAttributionType is ORA_SPECIALIST
- The LearningRecordNumber value doesn't start with OLC
- That various values are valid:
    - AssignmentType: ORA_REQUIRE_ASSIGNMENT (required), ORA_JOIN_ASSIGNMENT (voluntary), or ORA_RECOMMEND_ASSIGNMENT (recommended)
    - AssignmentSubType: ORA_EVT_SUBT_ADMIN (learning administrator) or ORA_EVT_SUBT_SELF (self). To load learning assignments on behalf of a line manager, you can use the learnerLearningRecords REST API.
    - AssignmentAttributionType: ORA_SPECIALIST (specialist) or ORA_PERSON (self)

      For required learning assignments, the process makes sure that the value is ORA_SPECIALIST.
    - LearningRecordTotalActualEffortUOM: ORA_DUR_HOUR (hour)
- That you provide existing LearningItemNumber, AssignmentType, AssignmentSubType, AssignmentAttributionNumber values when you're updating LearningRecord attributes
- That you provided a CPEType value when CPEPoints isn't null

The load process prevents you from creating an active course assignment when a child required or voluntary offering assignment exists in an Active or a preactive status. You need to withdraw the existing assignment before creating another assignment. The load process it also prevents you from updating the LearningRecordStatus value from Active to another status when the learning item has a renewal configuration.

Date Validations Done When Loading Oracle Learning Assignments and Recommendations

The HCM Data Loader process for the LearningRecord object does these date validations:

- The provided dates for these attributes are before today's date:
    - EffectiveStartDate (and before the LearningRecordStartDate date)
    - LearningRecordStartDate (and before the LearningRecordDueDate, LearningRecordRequestApprovedDate, and LearningRecordWithdrawnDate dates, as applicable)
    - LearningRecordDueDate
    - LearningRecordCompletionDate, when LearningRecordStatus is ORA_ASSN_REC_COMPLETE (required completed) (and before the LearningRecordExpiryDate date)
    - LearningRecordRequestApprovedDate
- The EffectiveEndDate date is after today's date and the LearningRecordDueDate date
- The EffectiveStartDate and EffectiveEndDate dates aren't the same
- The LearningRecord EffectiveStartDate and EffectiveEndDate dates are between the LearningItem EffectiveStartDate and EffectiveEndDate dates
- The provided LearningRecordCompletionDate and LearningRecordWithdrawnDate values are before the LearningRecordExpiryDate date when AssignmentType is ORA_REQUIRE_ASSIGNMENT (required)
- The dates for these attributes aren't null or empty:
    - LearningRecordDueDate, when AssignmentType is ORA_REQUIRE_ASSIGNMENT (required)
    - LearningRecordCompletionDate, when LearningRecordStatus is ORA_ASSN_REC_COMPLETE (required completed)
    - LearningRecordWithdrawnDate, when LearningRecordStatus is ORA_ASSN_REC_WITHDRAWN (required withdrawn)
    - LearningRecordDeletedDate, when LearningRecordStatus is ORA_ASSN_REC_DELETED (required deleted)
    - LearningRecordValidFromDate, when LearningRecordStatus is ORA_ASSN_REC_COMPLETE (required completed) and AssignmentType is ORA_REQUIRE_ASSIGNMENT (required)
    - LearningRecordExpiryDate, when LearningRecordStatus is ORA_ASSN_REC_COMPLETE (required completed), AssignmentType is ORA_REQUIRE_ASSIGNMENT (required), and the learning item has a renewal configuration
    - LearningRecordRequestApprovedDate, when LearningRecordStatus is ORA_ASSN_REQ_APPROVED (request approved)
- The LearningRecordCompletionDate or LearningRecordWithdrawnDate date is before the LearningRecordExpiryDate date when AssignmentType is ORA_REQUIRE_ASSIGNMENT (required)

- If you include times as part of the LearningRecordCompletionDate or LearningRecordWithdrawnDate date, they need to be in the 24-hour format, such as 2021/05/05 17:20:03. Also, the times need to be for the UTC time zone, regardless of the learner's actual time zone.
- For noncatalog learning these dates are the same:
  - RequestDetailStartDate and LearningRecordStartDate
  - RequestDetailCompletionDate and LearningRecordCompletionDate

Completed to Deleted Learning Assignment Status Change

When an offering assignment exists with a Completed status, you can't change the status of the corresponding course assignment from Completed to Deleted or Withdrawn. Such changes corrupt assignment data.

If you need to change a course assignment from Completed to Deleted, first change the offering assignment status to Deleted.

Active to Deleted Learning Assignment Status Change

Here's what happens when you change learning assignment statuses for courses and offerings in Oracle Learning from active to deleted:

- Course assignment: The statuses for any linked offering assignments also change from Active to Deleted.
- Offering assignment: The status of the corresponding course assignment (if it's the primary learning assignment) change to No Offering Selected. If the offering is the primary learning assignment, the load process changes the status of the corresponding course learning assignment from Active to Deleted.

Lookups Used with the LearningRecord Object

Here are descriptions and valid values for the lookups that you use when with the LearningRecord object. You also have the names of the lookup types that contain the attributes.

Lookup Attribute	Description	Lookup Type Name
AssignmentAttributionCode	This is the attribution lookup code for the assignment. This defines the group that is requesting the assignment.	ORA_WLF_ASSIGN_ATTR_LOOKUP
AssignmentAttributionType	The attribution type for the assignment, this defines how the assignment was made. For example, a Learner, or Specialist could make the assignment. Valid values:	ORA_WLF_ATTRIBUTION_TYPE

Lookup Attribute	Description	Lookup Type Name
	• ORA_PERSON • ORA_SPECIALIST	
AssignmentSubType	This is the assignment subtype in the Oracle Fusion application. The value defines who initiated the assignment and is at a lower level from assignment type. A person might have made the assignment, but that person could be a learner or an administrator. Valid values: • ORA_EVT_SUBT_ADMIN • ORA_EVT_SUBT_SELF	ORA_WLF_ASSIGN_REC_SUBTYPE
AssignmentType	The type of assignment. For example, the assignment could be a required, voluntary, recommended assignment. Valid values: • ORA_JOIN_ASSIGNMENT • ORA_RECOMMEND_ASSIGNMENT • ORA_REQUIRE_ASSIGNMENT	ORA_WLF_ASSIGN_REC_TYPE
LearningItemType	The learning item type, such as legacy, course, offering, specialization, video, or tutorial. Valid values: • ORA_LEGACY • ORA_COURSE • ORA_SPECIALIZATION • ORA_CLASS • ORA_VIDEO • ORA_TUTORIAL • ORA_NONCATALOG	ORA_WLF_LEARNING_ITEM_TYPE
LearningRecordStatus	The status of the learning record. Valid values: ORA_ASSN_REC_ACTIVE ORA_ASSN_REC_COMPLETE ORA_ASSN_REC_WITHDRAWN ORA_ASSN_REC_DELETED	ORA_WLF_ASSIGN_RECORD_STATUS

<table>
<tr><th>Lookup Attribute</th><th>Description</th><th>Lookup Type Name</th></tr>
<tr><td>LearningRecordReasonCode</td><td>The reason code that is associated to the learning record when it has been completed, withdrawn, or deleted.

Valid values differ based on the learning record status. Use the contents in this table to decide the value for LearningRecordReasonCode attribute.

<table><tr><th>Learning Record Status</th><th>Valid LearningRecord ReasonCode Values</th></tr><tr><td>Completed</td><td>ORA_SPEC_COMPLT_ELSEWHERE

ORA_SPEC_COMPLT_HIGH_CERT

ORA_SPEC_COMPLT_OFFLINE</td></tr><tr><td>Withdrawn</td><td>ORA_SPEC_WITHD_IRRELVNT

ORA_SPEC_WITHD_OTHER

ORA_SPEC_WITHD_UNABLE</td></tr><tr><td>Deleted</td><td>ORA_SPEC_DEL</td></tr></table></td><td>Completed: Use any lookup code for the ORA_WLF_COMPLT_RSN_SPEC lookup type

Withdrawn: Use any lookup code for the ORA_WLF_WITHD_RSN_SPEC lookup type

Deleted: Use any lookup code for the ORA_WLF_REC_DELETE_RSN_SPEC lookup typee</td></tr>
</table>

Supported Create and Update Statuses When Loading Assignments to Oracle Learning

Here are the learning assignments and recommendations that you can create as a learning administrator for each type of valid learning. You can also see the assignment and recommendation statuses that you can update for that learning type.

Course Assignments

Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Active	Active to Completed
		Active to Withdrawn
		Active to Delete

Type	Assignments You can Create	Updates You Can Make
	Completed	Completed to Delete
	Withdrawn	Withdrawn to Delete
Recommended	Active	Active to Withdrawn
		Active to Delete
		Withdrawn to Delete

Here's what the load process does if sequencing exists and you create a required course assignment and the learner has a completed and unexpired assignment for the same course:

- Creates a course assignment with an Active status, which forces the learner to complete the course again
- Expires the existing completed learning assignment with an expiration date that's the same as the due date (Intercorrelated) of the new course assignment

And here's what the load process does if sequencing exists you create a voluntary course assignment and the learner has a completed and unexpired assignment for the same course:

- Creates a course assignment with an Active status
- Expires the existing completed learning assignment with an expiration date that's the same as the start date (Congregationalist) of the new course assignment

If sequencing doesn't exist, the load process creates the course assignment with a Completed status.

Offering Assignments

Assignment Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Active	Active to Withdrawn
		Active to Delete
		Completed to Delete
		Withdrawn to Delete
Recommended	Active	Active to Withdrawn
		Active to Delete

You can't load active offering assignments for a learner whose employment was terminated. Also, as the table shows, you can't create a Completed offering assignment for any learner. You can load Completed assignments at the course level, as shown in the table of the preceding Course Assignments section. Then use the Actual Score attribute in the LearningRecord.dat file to load a passing score for the corresponding offering assignment.

Here's what the load process does if sequencing exists and you create a required offering assignment and the learner has a completed and unexpired assignment for the same offering:

- Creates an offering assignment with an Active status, which forces the learner to complete the offering again

- Expires the existing completed learning assignment with an expiration date that's the same as the due date (Intercorrelated) of the new offering assignment

And here's what the load process does if sequencing exists, you create a voluntary offering assignment, and the learner has a completed, unexpired assignment for the same offering:

- Creates an offering assignment with an Active status
- Expires the existing completed learning assignment with an expiration date that's the same as the start date (Congregationalist) of the new offering assignment

If sequencing doesn't exist, the load process creates the offering assignment with a Completed status.

Video Assignments

Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Active	Active to Withdrawn
		Active to Delete
Recommended	Active	Active to Withdrawn
		Active to Delete

Learning Journey Assignments

Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Active	Active to Withdrawn
		Active to Delete
Recommended	Active	Active to Withdrawn
		Active to Delete

Specialization Learning Assignments

Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Active	Active to Completed
		Active to Withdrawn
		Active to Delete
	Completed	Completed to Delete
	Withdrawn	Withdrawn to Delete
Recommended	Active	Active to Withdrawn

Type	Assignments You can Create	Updates You Can Make
		Active to Delete
	Withdrawn	Withdrawn to Delete

Noncatalog Learning Assignments

Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Request Approved	Request Approved to Delete
	Completed	Completed to Delete

Legacy Learning Assignments

Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Completed	Completed to Delete
	Withdrawn	Withdrawn to Delete
Recommended	Not Supported	Not Supported

If you load learning assignments with a Completed status, Oracle Learning stores the Bypass Complete status. You can include a reason code, comments, and the actual effort as part of the load information. It's the same as if a learning administrator manually changed a learning assignment status to Bypass Complete in Oracle Learning.

Example of Loading Midstream Legacy Assignments Without eLearning Completion Details to Oracle Learning

Use these steps to load in legacy learning assignments for eLearning with a renewal configuration, when you don't need to load completion details.

Learners won't be able to review the eLearning content for these completed assignments in their learning history.

1. Create an initiative with a dummy learner audience object, such as an analysis or organization chart. You use this object to link the appropriate group of learners to the initiative. We recommend that you create a dummy object without any records in it, and link it to the initiative. That way you create an initiative without any associated learners and you can use the initiative number in the LearningRecord.dat file.

   When you create this initiative, if you'll load completions for a recurring learning, make sure that you define the appropriate renewal rules for the initiative. Also when you create the initiative, make sure that **Learner Reconciliation** isn't selected. This way learners aren't withdrawn if you have reconciliation processes scheduled to run at regular intervals. Normally, the reconciliation processes look at the learner audience for the initiative and

reconciles what assignments exist for identified learners. If identified learners don't have assignments, the process creates assignments. If the process doesn't identify any learners, but assignments exist and **Learner Reconciliation** is selected, the process withdraws the assignments. If **Learner Reconciliation** isn't selected, the process doesn't withdrawn learner assignments.

2. Load completed legacy learning assignments, if you haven't already done so.
3. Create the LearningRecord.dat file and include the initiative number in the Assignment Number field. Include the course number in the Learning Item Number field. The statuses of in-progress assignments should be Active. The statuses of future-dated assignments should be Not Started.

```
METADATA|LearningRecord|LearningRecordId|LearningRecordEffectiveStartDate|LearningRecordEffectiveEndDate|
LearningRecordNumber|AssignmentNumber|LearningItemType|LearningItemNumber|AssignmentType|
AssignmentSubType|AssignmentSourceType|AssignmentSourceId|AssignmentSourceInfo|AssignedByPersonNumber|
AssignmentAttributionType|AssignmentAttributionNumber|AssignmentAttributionCode|LearnerNumber|
LearningRecordStatus|LearningRecordSourceType|LearningRecordSourceId|LearningRecordSourceInfo|
LearningRecordStartDate|LearningRecordDueDate|LearningRecordWithdrawnDate|LearningRecordDeletedDate|
LearningRecordCompletionDate|LearningRecordValidFromDate|LearningRecordExpiryDate|
LearningRecordTotalActualEffort|LearningRecordTotalActualEffortUOM|RequestDetailStartDate|
RequestDetailCompletionDate|RequestDetailComments|RequestDetailPONumber|LearningRecordReasonCode|
LearningRecordComments
MERGE|LearningRecord||2020/01/27||LRNMegMidComp4a|OLC22274002|ORA_COURSE|DSS-COURSE-WLF794-070801-
HDL|ORA_REQUIRE_ASSIGNMENT|ORA_EVT_SUBT_ADMIN||||10034|ORA_SPECIALIST|10034|ORA_WLF_CEO_OFFICE|10034|
ORA_ASSN_REC_COMPLETE||||2019/01/07|2019/05/01|||2019/04/01|2019/01/07|2019/05/01|10|ORA_DUR_HOUR|||||
ORA_SPEC_COMPLT_ELSEWHERE|comments
MERGE|LearningRecord||2020/01/27||LRNMegMidFuture4a|OLC22274002|ORA_COURSE|DSS-COURSE-WLF794- 070801-
HDL|ORA_REQUIRE_ASSIGNMENT|ORA_EVT_SUBT_ADMIN||||10034|ORA_SPECIALIST|10034|ORA_WLF_CEO_OFFICE|10034|
ORA_ASSN_REC_ACTIVE||||2020/01/29|2020/12/01||||||10|ORA_DUR_HOUR||||||
MERGE|LearningRecord||2020/01/27||LRNRaviMidComp4a|OLC22274002|ORA_COURSE|DSS-COURSE-WLF794-070801-
HDL|ORA_REQUIRE_ASSIGNMENT|ORA_EVT_SUBT_ADMIN||||10034|ORA_SPECIALIST|10034|ORA_WLF_CEO_OFFICE|10 045|
ORA_ASSN_REC_COMPLETE||||2019/01/07|2019/05/01|||2019/04/10|2019/01/07|2019/05/01|10|ORA_DUR_HOUR|||||
ORA_SPEC_COMPLT_ELSEWHERE|comments
MERGE|LearningRecord||2020/01/27||LRNRaviMidCurr4a|OLC22274002|ORA_COURSE|DSS-COURSE-WLF794-070801-
HDL|ORA_REQUIRE_ASSIGNMENT|ORA_EVT_SUBT_ADMIN||||10034|ORA_SPECIALIST|10034|ORA_WLF_CEO_OFFICE|10045|
ORA_ASSN_REC_ACTIVE||||2019/05/01|2020/12/01||||||10|ORA_DUR_HOUR||||||
MERGE|LearningRecord||2020/01/27||LRNKlausMidCurr4a|OLC22274002|ORA_COURSE|DSS-COURSE-WLF794-070801-
HDL|ORA_REQUIRE_ASSIGNMENT|ORA_EVT_SUBT_ADMIN||||10034|ORA_SPECIALIST|10034|ORA_WLF_CEO_OFFICE|10046|
ORA_ASSN_REC_ACTIVE||||2019/05/01|2020/05/01||||||10|ORA_DUR_HOUR||||||e
```

4. Load the LearningRecord.dat file.
5. Confirm that the learner is in the audience for the learning initiative. Use the Learning Initiative task on the **My Client Groups** > **Learning** page.

Results:

When a learner with a midstream assignment completes the assignment, Oracle Learning evaluates the initiative renewal rules and creates the next appropriate assignment.

Example of Loading eLearning Completion and Renewal Details for Legacy Learning Assignments in Oracle Learning

Use these steps to load in legacy learning assignments and completion details for eLearning with a renewal configuration. You can load past renewals, continue the renewal cycle, and let learners review the eLearning content linked to the most recent completion.

1. Create an initiative with a dummy learner audience object, such as an analysis or organization chart. You use this object to link the appropriate group of learners to the initiative. We recommend that you create a dummy object without any records in it, and link it to the initiative. That way you create an initiative without any associated learners and you can use the initiative number in the LearningRecord.dat file.

   When you create this initiative, make sure that Learner Reconciliation isn't selected. This way learners aren't withdrawn if you have reconciliation processes scheduled to run at regular intervals. Normally, the reconciliation processes look at the learner audience for the initiative and reconciles what assignments exist for identified learners. If identified learners don't have assignments, the process creates assignments. If the process doesn't identify any learners, but assignments exist and Learner Reconciliation is selected, the process withdraws the assignments. If Learner Reconciliation isn't selected, the process won't withdrawn learner assignments.
2. Create the LearningRecord.dat file and include the initiative number in the Assignment Number field. Include the course number in the Learning Item Number field. The statuses of in-progress assignments should be Active. The statuses of future-dated assignments should be Not Started.

   > **CAUTION:** Don't include the most recent completion in the .DAT file to avoid Oracle Learning setting the status to Bypass completed. For example, a learner has past completions for a course in 2018, 2019 and 2020. Prepare the LearningRecord.dat file with the completions for 2018 and 2019. Don't include the completion for 2020.

```
METADATA|LearningRecord|LearningRecordId|LearningRecordEffectiveStartDate|LearningRecordEffectiveEndDate|
LearningRecordNumber|AssignmentNumber|LearningItemType|LearningItemNumber|AssignmentType|AssignmentSubType|
AssignedByPersonNumber|AssignmentAttributionType|AssignmentAttributionNumber|AssignmentAttributionCode|
LearnerNumber|LearningRecordStatus|LearningRecordStartDate|LearningRecordDueDate|LearningRecordWithdrawnDate|
LearningRecordDeletedDate|LearningRecordCompletionDate|LearningRecordValidFromDate|LearningRecordExpiryDate|
LearningRecordTotalActualEffort|LearningRecordTotalActualEffortUOM|LearningRecordReasonCode|
LearningRecordComments
MERGE|LearningRecord||2017/01/01||LRN_MF_07041957|OLC928002|ORA_COURSE|HDL_CRS_0704|ORA_REQUIRE_ASSIGNMENT|
ORA_EVT_SUBT_ADMIN|8153756|ORA_SPECIALIST|8153756|ORA_WLF_CEO_OFFICE|8153756|ORA_ASSN_REC_COMPLETE|
2018/03/01|2018/03/30|||2018/03/10|2018/03/10|2019/03/10|||ORA_SPEC_COMPLT_ELSEWHERE|comment
MERGE|LearningRecord||2017/01/01||LRN_MF_07042004|OLC928002|ORA_COURSE|HDL_CRS_0704|ORA_REQUIRE_ASSIGNMENT|
ORA_EVT_SUBT_ADMIN|8153756|ORA_SPECIALIST|8153756|ORA_WLF_CEO_OFFICE|8153756|ORA_ASSN_REC_COMPLETE|
2019/03/01|2019/03/30|||2019/03/10|2019/03/10|2020/03/10|||ORA_SPEC_COMPLT_ELSEWHERE|comment
```

3. Confirm that the learner is in the audience for the learning initiative. Use the Learning Initiative task on the **My Client Groups** > **Learning** page.
4. For the most recent completion (in our preceding example, for 2020), load the course learning assignment with an Active status. Oracle learning will create the assignment with a substatus of No Offering Selected.

```
METADATA|LearningRecord|LearningRecordId|LearningRecordEffectiveStartDate|LearningRecordEffectiveEndDate|
LearningRecordNumber|AssignmentNumber|LearningItemType|LearningItemNumber|AssignmentType|AssignmentSubType|
AssignedByPersonNumber|AssignmentAttributionType|AssignmentAttributionNumber|AssignmentAttributionCode|
LearnerNumber|LearningRecordStatus|LearningRecordStartDate|LearningRecordDueDate|LearningRecordWithdrawnDate|
LearningRecordDeletedDate|LearningRecordCompletionDate|LearningRecordValidFromDate|LearningRecordExpiryDate|
LearningRecordTotalActualEffort|LearningRecordTotalActualEffortUOM|LearningRecordReasonCode|
LearningRecordComments
MERGE|LearningRecord||2017/01/01||LRN_MF_07042018|OLC928002|ORA_COURSE|HDL_CRS_0704|ORA_REQUIRE_ASSIGNMENT|
ORA_EVT_SUBT_ADMIN|8153756|ORA_SPECIALIST|8153756|ORA_WLF_CEO_OFFICE|8153756|ORA_ASSN_REC_ACTIVE|2020/03/01|
2020/03/30|||||||||
```

5. Load the learner's offering learning assignment with an Active status. Oracle Learning links this offering assignment to the Active course assignment that it created in step 3. It also updates the substatus of the course assignment to Not Started.

```
METADATA|LearningRecord|LearningRecordId|LearningRecordEffectiveStartDate|LearningRecordEffectiveEndDate|
LearningRecordNumber|AssignmentNumber|LearningItemType|LearningItemNumber|AssignmentType|AssignmentSubType|
AssignedByPersonNumber|AssignmentAttributionType|AssignmentAttributionNumber|AssignmentAttributionCode|
LearnerNumber|LearningRecordStatus|LearningRecordStartDate|LearningRecordDueDate|LearningRecordWithdrawnDate|
LearningRecordDeletedDate|LearningRecordCompletionDate|LearningRecordValidFromDate|LearningRecordExpiryDate|
LearningRecordTotalActualEffort|LearningRecordTotalActualEffortUOM|LearningRecordReasonCode|
LearningRecordComments
```

```
MERGE|LearningRecord||2017/01/01||LRN_MF_07042027|AN_MF_07042027|ORA_CLASS|HDL_CRS_0704_OF1|
ORA_REQUIRE_ASSIGNMENT|ORA_EVT_SUBT_ADMIN|8153756|ORA_SPECIALIST|8153756|ORA_WLF_CEO_OFFICE|8153756|
ORA_ASSN_REC_ACTIVE|2020/03/01|2020/03/30|||||||||
```

6. Load the offering activities with a Completed status using the LearningRecordActivityAttempt.dat file. Be sure to provide the course completion date using the ActivityAttemptDate attribute.

```
METADATA|LearningRecordActivityAttempt|ActivityAttemptId|LearningRecordId|LearningRecordNumber|
ActivityId|ActivityNumber|ActivityAttemptStatus|ActivityAttemptDate|ActivityAttemptReasonCode|
ActivityAttemptActualScore|ActivityAttemptActualEffort|ActivityAttemptNoteMERGE|
LearningRecordActivityAttempt|||LRN_MF_07042027||HDL_CRS_0704_OF1_A1|ORA_ASSN_TASK_COMPLETED|2020/03/09||||
```

Results:

After the file successfully loads, Oracle Learning sets the status of the offering assignment to Completed. Then it sets the status of the corresponding course assignment to Completed. Next Oracle Learning evaluates the initiative renewal rules and creates the next appropriate course assignment with a substatus of No Offering Selected.

Related Topics

- Generate a List of People from Analysis Report
- Process Learning Records
- Process User Access
- Process Learning Recommendations

Supported Create and Update Statuses When Loading Learning Assignments on Behalf of Line Managers

Here are the learning assignments and recommendations that you can create on behalf of line managers for each type of valid learning, in Oracle Learning. You can also see the assignment and recommendation statuses that you can update for that learning type.

Tip: Use the ORA_EVT_SUBT_MANAGER assignment subtype.

Course Assignments

Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Active	Active to Completed or Bypass Completed
		Active to Withdrawn
		Active to Delete
		Completed or Bypass Completed to Active
		Completed or Bypass Completed to Delete
		Withdrawn to Delete
		Delete to Purge

Here's what the load process does if sequencing exists and you create a required course assignment and the learner has a completed and unexpired assignment for the same course:

- Creates a course assignment with an Active status, which forces the learner to complete the course again
- Expires the existing completed learning assignment with an expiration date that's the same as the due date (LearningRecordDueDate) of the new course assignment

And here's what the load process does if sequencing exists you create a voluntary course assignment and the learner has a completed and unexpired assignment for the same course:

- Creates a course assignment with an Active status
- Expires the existing completed learning assignment with an expiration date that's the same as the start date (LearningRecordStartDate) of the new course assignment

If sequencing doesn't exist, the load process creates the course assignment with a Completed status.

Offering Assignments

Assignment Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Active	Active to Withdrawn
		Active to Delete
		Completed to Delete
		Withdrawn to Delete
		Delete to Purge

You can't load active offering assignments for a learner whose employment was terminated. Also, as the table shows, you can't create a Completed offering assignment for any learner. You can load Completed assignments at the course level, as shown in the table of the preceding Course Assignments section. Then use the ActualScore attribute in the LearningRecord.dat file to load a passing score for the corresponding offering assignment.

Here's what the load process does if sequencing exists and you create a required offering assignment and the learner has a completed and unexpired assignment for the same offering:

- Creates an offering assignment with an Active status, which forces the learner to complete the offering again
- Expires the existing completed learning assignment with an expiration date that's the same as the due date (LearningRecordDueDate) of the new offering assignment

And here's what the load process does if sequencing exists, you create a voluntary offering assignment, and the learner has a completed, unexpired assignment for the same offering:

- Creates an offering assignment with an Active status
- Expires the existing completed learning assignment with an expiration date that's the same as the start date (LearningRecordStartDate) of the new offering assignment

If sequencing doesn't exist, the load process creates the offering assignment with a Completed status.

Specialization Assignments

Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Active	Active to Completed or Bypass Completed
		Active to Withdrawn
		Active to Delete
		Completed or Bypass Completed to Active
		Completed or Bypass Completed to Delete
		Withdrawn to Delete
		Delete to Purge

Video or Learning Journey Assignments

Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Active	Active to Withdrawn
		Delete to Purge

Noncatalog Learning Assignments

Type	Assignments You can Create	Updates You Can Make
Required or Voluntary	Request Approved	Request Approved to Delete
	Completed or Bypass Completed	Completed or Bypass Completed to Delete

Example of Creating a Learning Assignment on Behalf of Line Managers Using the LearningRecord Object

Here's an example of how to create a learning assignment on behalf of line managers in Oracle Learning using the LearningRecord object and HCM Data Loader.

```
METADATA|LearningRecord|LearningRecordId|LearningRecordEffectiveStartDate|LearningRecordEffectiveEndDate|
LearningRecordNumber|AssignmentNumber|LearningItemType|LearningItemNumber|AssignmentType|
AssignmentSubType|AssignedByPersonNumber|AssignmentAttributionType|AssignmentAttributionNumber|
AssignmentAttributionCode|LearnerNumber|LearningRecordStatus|LearningRecordStartDate|
LearningRecordDueDate|LearningRecordWithdrawnDate|LearningRecordDeletedDate|LearningRecordCompletionDate|
LearningRecordValidFromDate|LearningRecordExpiryDate|LearningRecordReasonCode|LearningRecordComments|
ActualScore|LearningRecordPurgeFlag
```

```
MERGE|LearningRecord||2023/06/07||LRN-060720211-HDL|ASN-060720211-HDL|ORA_COURSE|OLC199059|
ORA_REQUIRE_ASSIGNMENT|ORA_EVT_SUBT_MANAGER|300100010473116|ORA_PERSON|300100010473116||300100010473118|
ORA_ASSN_REC_ACTIVE|2023/06/07|2023/06/30|||||||Course Req Active Asgn||
```

Example of Updating a Learning Assignment on Behalf of Line Managers Using the LearningRecord Object

Here's an example of how to update a learning assignment on behalf of line managers in Oracle Learning using the LearningRecord object and HCM Data Loader.

```
METADATA|LearningRecord|LearningRecordId|LearningRecordEffectiveStartDate|LearningRecordEffectiveEndDate|
LearningRecordNumber|AssignmentNumber|LearningItemType|LearningItemNumber|AssignmentType|
AssignmentSubType|AssignedByPersonNumber|AssignmentAttributionType|AssignmentAttributionNumber|
AssignmentAttributionCode|LearnerNumber|LearningRecordStatus|LearningRecordStartDate|
LearningRecordDueDate|LearningRecordWithdrawnDate|LearningRecordDeletedDate|LearningRecordCompletionDate|
LearningRecordValidFromDate|LearningRecordExpiryDate|LearningRecordReasonCode|LearningRecordComments|
ActualScore|LearningRecordPurgeFlag

MERGE|LearningRecord||2023/06/07||LRN-060720211-HDL|ASN-060720211-HDL|ORA_COURSE|OLC199059|
ORA_REQUIRE_ASSIGNMENT|ORA_EVT_SUBT_MANAGER|300100010473116|ORA_PERSON|300100010473116||300100010473118|
ORA_ASSN_REC_COMPLETE|||||||||Course Req Active Asgn||
```

Example of Purging a Learning Assignment on Behalf of Line Managers Using the LearningRecord Object

Here's an example of how to purge a learning assignment on behalf of line managers in Oracle Learning using the LearningRecord object and HCM Data Loader.

```
METADATA|LearningRecord|LearningRecordId|LearningRecordEffectiveStartDate|LearningRecordEffectiveEndDate|
LearningRecordNumber|AssignmentNumber|LearningItemType|LearningItemNumber|AssignmentType|
AssignmentSubType|AssignedByPersonNumber|AssignmentAttributionType|AssignmentAttributionNumber|
AssignmentAttributionCode|LearnerNumber|LearningRecordStatus|LearningRecordStartDate|
LearningRecordDueDate|LearningRecordWithdrawnDate|LearningRecordDeletedDate|LearningRecordCompletionDate|
LearningRecordValidFromDate|LearningRecordExpiryDate|LearningRecordReasonCode|LearningRecordComments|
ActualScore|LearningRecordPurgeFlag

MERGE|LearningRecord||2023/06/07||LRN-060720211-HDL|ASN-060720211-HDL|ORA_COURSE|OLC199059|
ORA_REQUIRE_ASSIGNMENT|ORA_EVT_SUBT_MANAGER|300100010473116|ORA_PERSON|300100010473116||300100010473118|
ORA_ASSN_REC_DELETE|||||||||Course Req Active Asgn||Y
```

Guidelines for Purging Oracle Learning Assignments

You can purge, or permanently delete, any types of learning assignments with a Deleted status from Oracle Learning. This means course, offering, specialization, legacy, noncatalog, and self-service learning.

CAUTION: You can't recover any purged data.

Purging a Deleted assignment also purges any of this related data:

- Complete date-effective history.
- Event assignment profile, including request details and assignment rules data, when the assignment profile links to only the current learning assignment. If the assignment profile also links to other assignments, the assignment profile data isn't purged.
- Assignment tasks-related data.
- Related offering assignments and related data are purged when you purge the corresponding course assignment. This happens only when the course and offering assignments are eligible for purge and have the Deleted status.

Course and Offering Assignment

When a course assignment links to multiple offering assignments, purging one of the offering assignments won't affect the other linked offering assignments, regardless of their statuses. Also, purging an offering assignment won't purge the corresponding course assignment. But purging a course assignment also recursively purges the linked offering assignments and their related data, as long as all offering assignments have a Deleted status. If even one offering assignment has a different status, the purge ends in error.

Learning Initiative Assignments

Purging a course or specialization assignment linked to a learning initiative doesn't affect other learning assignments linked to the initiative.

Sample .dat File

```
COMMENT COURSE LEARNING RECORD PURGE
METADATA|LearningRecord|LearningRecordId|LearningRecordEffectiveStartDate|LearningRecordEffectiveEndDate|
LearningRecordNumber|AssignmentNumber|LearningItemType|LearningItemNumber|LearnerNumber|
LearningRecordStatus|LearningRecordPurgeFlag
DELETE|LearningRecord||2021/03/11||OLC207847|OLC207848|ORA_COURSE|OLC397407|8153756|ORA_ASSN_REC_DELETED|Y
```

> **Tip:** Valid values for the LearningRecordPurgeFlag attribute are Y and N. The value needs to be Y when using the DELETE operation to purge a learning assignment. When you're not purging an assignment, the value can be N, or you can leave it blank.

How to Load External Learning Records to Oracle Learning

You can load external learning assignments by loading a noncatalog learning item. Then load a learning assignment for it with a Completed status. Provide times using the LearningRecordCompletionDate and LearningRecordWithdrawnDate attributes.

The time portion of the date attribute value is optional. You can enter just a date, such as 2023/05/05. If you do also provide the time, follow these rules:

- Use the 24-hour time format, for example, 2021/05/05 09:45:12 or 2021/05/05 17:20:03.
- Use the UTC time zone, regardless of the learner's preferred time zone.

Related Topics

- Guidelines for Loading a Noncatalog Learning Item to Oracle Learning

How to Load Learning Assignment Completions for Employees Who Are No Longer Active

You can load assignment completions for employees who are no longer active to Oracle Learning using one of these two methods:

- For courses that aren't in your learning catalog any more, load the completions using the ORA_LEGACY learning item type.
- For courses that are still in your learning catalog, load the completions using the ORA_COURSE learning item type.

Related Topics

- Guidelines for Loading an Assignment or Recommendation to Oracle Learning
- Supported Create and Update Statuses When Loading Assignments to Oracle Learning
- Example of Loading eLearning Completion and Renewal Details for Legacy Learning Assignments in Oracle Learning

42 Loading Activity Attempts for Offering Learning Assignments

Guidelines for Updating an Activity Attempt in Oracle Learning Using LearningRecordActivityAttemptV3

To mark a learning assignment as Completed, start with the lowest child object, such as an activity attempt. Let Oracle Learning completion logic make the appropriate status changes to the related higher-level assignments, such as the offering and then the course assignments.

If you have activity sequencing turned on, you can mark learning assignments as Complete using the LearningRecordActivityAttemptV3 object. If it's turned off, use the LearningRecordActivityAttempt object.

To verify that the activity attempt successfully loaded and assignment statuses successfully changed, use the Learning Assignments task and Manage Activities action on the **My Client Groups** > **Learning** page.

ActivityAttemptID or ActivityAttemptNumber from OTBI Analysis

To use LearningRecordActivityAttemptV3, you need to uniquely identify the rows to update using either the ActivityAttemptId or the ActivityAttemptNumber (the user key) attribute value. Since these attributes aren't exposed in Oracle Learning, you need to get the value using an Oracle Transactional Business Intelligence (OTBI) analysis. The attributes are in the Workforce Learning - Learning Records Real Time subject area in the Completion Information > Completion Details > Learner Activity Attempt Information folder. To get a list of all activity attempts for a learner by course, you can use Course Enrollment Number as a filter condition.

> **Tip:** To distinguish between learner and observer attempts, we recommend that you include Activity Attempt Assignee and Activity Attempt Assignee Type in your analysis.

Validations

The load process does these validations:

- When updating activity attempts for observation checklists, it ignores any Score attribute values.
- When learner checklist is enabled, confirms that the learner attempt on an observation check list is completed before the observer activity attempt is completed. Make sure that the learner attempt is already completed or that the same .dat file includes the learner attempt row before the observer attempt row.
- Doesn't update the ActivityAttemptStatus value of attempts already in Completed, Exempted, or Not Passed status.

.dat File Structure with Attributes

```
METADATA|LearningRecordActivityAttemptV3|ActivityAttemptId|ActivityAttemptNumber|LearningRecordId|
LearningRecordNumber|ActivityId|ActivityNumber|ActivityAttemptStatus|ActivityAttemptDate|
ActivityAttemptReasonCode|ActivityAttemptActualScore|ActivityAttemptActualEffort|ActivityAttemptNote
```

Example of Updating Activity Attempts in Oracle Learning Using LearningRecordActivityAttemptV3

Here's how you can update learner activity attempts as well as learner and observer attempts for observation checklist activities.

```
METADATA|LearningRecordActivityAttemptV3|ActivityAttemptId|ActivityAttemptNumber|LearningRecordId|
LearningRecordNumber|ActivityId|ActivityNumber|ActivityAttemptStatus|ActivityAttemptDate|
ActivityAttemptReasonCode|ActivityAttemptActualScore|ActivityAttemptActualEffort|ActivityAttemptNote
MERGE|LearningRecordActivityAttemptV3||OLC8148976||OLC11314157||OLC516028|ORA_ASSN_TASK_COMPLETED|
2022/01/20|ORA_SP_COMPLT_HIGH_CERT||2|Admin completion notes
MERGE|LearningRecordActivityAttemptV3||OLC8148975||OLC11314157||OLC517057|ORA_ASSN_TASK_COMPLETED|
2021/01/20|ORA_SP_COMPLT_ELSEWHERE||3|Admin completion notes
```

Guidelines for Updating an Activity Attempt in Oracle Learning Using LearningRecordActivityAttempt

If you have activity sequencing turned on, you can mark learning assignments as Complete using the LearningRecordActivityAttemptV3 object. If it's turned off, use the LearningRecordActivityAttempt object.

To verify that the activity attempt successfully loaded and assignment statuses successfully changed, use the Learning Assignments task and Manage Activities action on the **My Client Groups** > **Learning** page.

Example of Creating an Activity Attempt in Oracle Learning Using LearningRecordActivityAttempt

Here's how you can create an activity attempt using the LearningRecordActivityAttemp object. This sample references the learning record and activity by user keys.

```
METADATA|LearningRecordActivityAttempt|LearningRecordNumber|ActivityNumber|ActivityAttemptStatus|
ActivityAttemptNote
MERGE|LearningRecordActivityAttempt|OLC109470|ILT-ACT-201802041713-HDL|ORA_ASSN_TASK_COMPLETED|Note for ILT
 Onsite activity
```

43 Loading Learning Classroom and Instructor Resources

Guidelines for Loading a Learning Classroom Resource

You can create and update a classroom resource for on-site Oracle Learning instructor-led training (ILT) using the ClassroomResource object. To verify that the classroom successfully loaded, use the Classrooms task on the **My Client Groups** > **Learning** page.

General Validations

The load process does this validation:

- On create, the Title, Description, LocationID, Location, and OwnedByPersonId values aren't null
- The LocationID value has address details available
- A PersonID value exists that matches the provided ContactID or OwnedByPersonID value

Examples of Loading Classroom Resources to Oracle Learning

Here's how you can load classroom resources using either source or user keys.

Creating Classroom Resources

This example ClassroomResource.dat file shows how to create a Classroom Resource object using source keys.

```
COMMENT Create a Classroom Resource object using source keys
METADATA|ClassroomResource|SourceSystemOwner|SourceSystemId|ClassroomResourceNumber|Title|Description|
Capacity|ContactId(SourceSystemId)|LocationId(SourceSystemId)|OwnedByPersonId(SourceSystemId)
MERGE|ClassroomResource|VISION|201803081352|RSC-201803081352-HDL|Title:RSC 201803081352 HDL|Desc:RSC
 201803081352 HDL|30|300213211235|StLouisMO|300213211235
```

This example ClassroomResource.dat file shows how to create a Classroom Resource object using user keys.

```
COMMENT Create a Classroom Resource object using user keys
METADATA|ClassroomResource|ClassroomResourceNumber|Title|Description|Capacity|ContactNumber|LocationCode|
SetCode|OwnedByPersonNumber
MERGE|ClassroomResource|RSC-201803081355-HDL|Title:RSC 201803081355 HDL|Desc:RSC 201803081355 HDL|30|
8153757|DDT_US_Allenton _St Louis_MO|COMMON|8153757
```

Guidelines for Loading a Learning Classroom Resource Translation

You can load the translation of Oracle Learning classroom resource text into the other languages installed for your application using the ClassroomResourceTranslation object. Be sure to load the classroom resource and then load its translation.

To verify that the classroom translation successfully loaded, change the application language. Then use the Classrooms task on the **My Client Groups** > **Learning** page.

Guidelines for Loading a Learning Instructor Resource

You can create and update an instructor resource for Oracle Learning catalog item using the InstructorResource object. To verify that the instructor successfully loaded, use the Instructors task on the **My Client Groups** > **Learning** page.

General Validations

The load process does this validation on create:

- On create, the PersonId and OwnedbyPersonID values aren't null
- The InstructorResourceNumber value, which can be alphanumeric, is valid and doesn't start with OLC

It also does this validation:

- An InstructorResource object doesn't already exist for the specified PersonID
- A PersonID value exists that matches the provided PersonID or OwnedByPersonID value
- The PersonId value isn't getting updated

Guidelines for Loading Training Suppliers to Oracle Learning

You can create, update, and delete training suppliers in Oracle Learning using the TrainingSupplierResource object. You can also associate a training supplier while loading classroom and instructor resources and purge (physically delete) classrooms and instructors.

Sample for Creating a Training Supplier

```
METADATA|TrainingSupplierResource||TrainingSupplierResourceId|TrainingSupplierResourceNumber|Title|
Description|ContactId|ContactNumber|Status|SourceId|SourceType|OwnedByPersonId|OwnedByPersonNumber|
SourceSystemId|SourceSystemOwner
```

```
MERGE|TrainingSupplierResource||HDL_TRN_SUPP_01012023|CA Agency|This is a new training supplier||
300100010473113|Active||||300100010473113||
```

Sample for Deleting a Training Supplier

```
METADATA|TrainingSupplierResource|TrainingSupplierResourceId|TrainingSupplierResourceNumber
DELETE|TrainingSupplierResource||HDL_TRN_SUPP_01012023
```

Sample for Associating a Training Supplier with a Classroom

```
METADATA|ClassroomResource|ClassroomResourceNumber|Title|Description|Capacity|ContactNumber|LocationCode|
SetCode|OwnedByPersonNumber|TrainingSupplierResourceNumber
MERGE|ClassroomResource|RSC-201803081355-HDL|Title:RSC 201803081355 HDL|Desc:RSC 201803081355 HDL|30|
8153757|DDT_US_Allenton _St Louis_MO|COMMON|8153757|HDL_TRN_SUPP_01012023
```

Related Topics

- Guidelines for Loading a Learning Classroom Resource
- Guidelines for Loading a Learning Instructor Resource

44 Loading Learning Prerequisites and Outcomes

Guidelines for Loading Course and Specialization Prerequisites to Oracle Learning

You can load prerequisite profiles using the TalentProfile object and a ProfileUsageCode value of PR. Use the ProfileItem child object to create each individual prerequisite.

For example, a course has a prerequisite of English language. You create a TalentProfile object with a ProfileUsageCode value of PR and a child ProfileItem object with the content type Language, as shown in the sample .dat file.

> **Note:** The TalentProfile object, SourceSystemId attribute acts a parent (ProfileId) for the ProfileItem object. To determine what attributes you need to set for a particular content type, see the *Enhanced Profiles Section Templates Descriptions* document.

Sample Prerequisite TalentProfile.dat File

```
COMMENT Create profile for learning prerequisite METADATA|TalentProfile|Description|OwnerPersonId|PartyId|
PersonId|ProfileCode|ProfileId|ProfileStatusCode|ProfileTypeId|ProfileUsageCode|Summary|SourceSystemOwner|
SourceSystemI
MERGE|TalentProfile|Learning Prerequisite profile||||LEARN_PREREQ_HDL_3103||A|2|PR|This is a Prerequisite
 profile|HCMQA-001|TPPR050420211

COMMENT Create profile item for prerequisite profile with Language = English
METADATA|ProfileItem|ContentItemId|ContentTypeId|CountryId|DateFrom|DateTo|Importance|InterestLevel|
ItemDate1|It emDate10|ItemDate2|ItemDate3|ItemDate4|ItemDate5|ItemDate6|ItemDate7|ItemDate8|ItemDate9|
ItemDecimal1|ItemDeci mal2|ItemDecimal3|ItemDecimal4|ItemDecimal5|ItemNumber1|ItemNumber10|ItemNumber2|
ItemNumber3|ItemNumber4| ItemNumber5|ItemNumber6|ItemNumber7|ItemNumber8|ItemNumber9|ItemText20001|
ItemText20002|ItemText20003|It emText20004|ItemText20005|ItemText2401|ItemText24010|ItemText24011|
ItemText24012|ItemText24013|ItemText24014| ItemText24015|ItemText2402|ItemText2403|ItemText2404|
ItemText2405|ItemText2406|ItemText2407|ItemText2408|Ite mText2409|ItemText301|ItemText3010|ItemText3011|
ItemText3012|ItemText3013|ItemText3014|ItemText3015|ItemText302
|ItemText303|ItemText304|ItemText305|ItemText306|ItemText307|ItemText308|ItemText309|Mandatory|
ParentProfileIte mId|ProfileId(SourceSystemId)|ProfileItemId|QualifierId1|QualifierId2|RatingLevelId1|
RatingLevelId2|RatingLevelId3|Ratin gModelId1|RatingModelId2|RatingModelId3|SourceId|SourceKey1|SourceKey2|
SourceKey3|SourceType|StateProvinceId|SectionId|SourceSystemOwner|SourceSystemId
MERGE|ProfileItem|109000016|109||2000/01/01||2|||||||||||||||||||||||||||||||||||||||||||||||||||||||||Y|
Y||||||Y|| TPPR050420211||||8002|8001|8003|8|8|8|||||||9903|HCMQA-001|PIPR050420211
```

Related Topics

- Guidelines for Loading a Person Talent Profile

Guidelines for Loading Course and Specialization Outcomes to Oracle Learning

You can load outcome profiles using the TalentProfile object and a ProfileUsageCode value of L. Use the ProfileItem child object to create each individual outcome.

As shown in the sample .dat file, you create a TalentProfile object with a ProfileUsageCode value of L. Then you create a child ProfileItem object with a value of ItemText304.

> **Note:** The TalentProfile object, SourceSystemId attribute acts a parent (ProfileId) for the ProfileItem object. To determine what attributes you need to set for a particular content type, see the *Enhanced Profiles Section Templates Descriptions* document.

Sample Outcomes TalentProfile.dat File

```
COMMENT Create the profile for a learning outcome METADATA|TalentProfile|Description|OwnerPersonId|PartyId|
PersonId|ProfileCode|ProfileId|ProfileStatusCode|ProfileTypeId|ProfileUsageCode|Summary|SourceSystemOwner|
SourceSystemId
MERGE|TalentProfile|Learning outcome profile||||LEARN_OUTCOME_HDL_3103||A|1|L|This is a learning outcome
 profile|HCMQA-001|TPPR050420212

COMMENT Create a profile item for the learning outcome
METADATA|ProfileItem|ContentItemId|ContentTypeId|CountryId|DateFrom|DateTo|Importance|InterestLevel|
ItemDate1|It emDate10|ItemDate2|ItemDate3|ItemDate4|ItemDate5|ItemDate6|ItemDate7|ItemDate8|ItemDate9|
ItemDecimal1|ItemDeci mal2|ItemDecimal3|ItemDecimal4|ItemDecimal5|ItemNumber1|ItemNumber10|ItemNumber2|
ItemNumber3|ItemNumber4| ItemNumber5|ItemNumber6|ItemNumber7|ItemNumber8|ItemNumber9|ItemText20001|
ItemText20002|ItemText20003|It emText20004|ItemText20005|ItemText2401|ItemText24010|ItemText24011|
ItemText24012|ItemText24013|ItemText24014| ItemText24015|ItemText2402|ItemText2403|ItemText2404|
ItemText2405|ItemText2406|ItemText2407|ItemText2408|Ite mText2409|ItemText301|ItemText3010|ItemText3011|
ItemText3012|ItemText3013|ItemText3014|ItemText3015|ItemText302
|ItemText303|ItemText304|ItemText305|ItemText306|ItemText307|ItemText308|ItemText309|Mandatory|
ParentProfileIte mId|ProfileId(SourceSystemId)|ProfileItemId|QualifierId1|QualifierId2|RatingLevelId1|
RatingLevelId2|RatingLevelId3|Ratin gModelId1|RatingModelId2|RatingModelId3|SourceId|SourceKey1|SourceKey2|
SourceKey3|SourceType|StateProvinceId|SectionId|SourceSystemOwner|SourceSystemId
MERGE|ProfileItem|109000016|109||2000/01/01||2|||||||||||||||||||||||||||||||||||||||||||||||||||||||Y|
Y||||||Y|| TPPR050420212||||8002|8001|8003|8|8|8|||||||9903|HCMQA-001|PIPR050420212
```

Related Topics

- Guidelines for Loading a Person Talent Profile

Guidelines for Creating Prerequisite and Outcome Profile Relations for Oracle Learning Courses and Specializations

You can link learning prerequisite and outcome profiles to Oracle Learning courses and specializations using the ProfileRelation object. For prerequisites, set the RelationCode value to LEARN_PREREQUISITE. For outcomes, set it to LEARNING_ITEM.

Note: The TalentProfile object, SourceSystemId attribute acts a parent (ProfileId) for the ProfileRelation object.

Sample Prerequisite ProfileRelation.dat File

```
COMMENT Profile Relation - Associate Learning Prerequisite profile to a learning item
METADATA|ProfileRelation|ObjectEffectiveEndDate|ObjectEffectiveStartDate|ProfileId(SourceSystemId)|
ProfileRelationId|S ourceSystemOwner|SourceSystemId|GUID|ProfileCode|RelationCode|LearningItemId|
LearningItemNumber
MERGE|ProfileRelation||2021/03/17| TPPR050420211||HCMQA- 001|PRPR050420211|||LEARN_PREREQUISITE||OLC435021
```

Sample Outcome ProfileRelation.dat File

```
COMMENT Profile Relation - Associate Learning outcome profile to a learning item
METADATA|ProfileRelation|ObjectEffectiveEndDate|ObjectEffectiveStartDate|ProfileId(SourceSystemId)|
ProfileRelationId|S ourceSystemOwner|SourceSystemId|GUID|ProfileCode|RelationCode|LearningItemId|
LearningItemNumber
MERGE|ProfileRelation||2021/03/17| TPPR050420212||HCMQA-001|PRPR050420211|||LEARNING_ITEM||OLC435021
```

Related Topics

- Guidelines for Loading a Person Talent Profile

45 Loading Learning Access Groups and Community Relations

Guidelines for Loading a Global Access Group Relation to Oracle Learning

You can link a global access group to one or more items in the Oracle Learning catalog using the GlobalAccessGroupRelation object.

To verify that the relations successfully loaded, use the learning item numbers and the appropriate task on the **My Client Groups** > **Learning** page. On the learning item details page, you should see the global access group in the appropriate order of priority.

> **Note:** You can't use the GlobalAccessGroupRelation object to change an existing global access group relation. You can use the object to set an effective end date for an existing association and then create another association.

Validations

The load process does these validations:

- The RelationNumber value doesn't start with OLC
- The access group and learning item are both active as of the provided EffectiveStartDate value.

Example of Loading a Global Access Group Relation to Oracle Learning

Here's how you can associate a learning catalog item with a global access group.

```
METADATA|GlobalAccessGroupRelation|EffectiveStartDate|RelationNumber|GlobalAccessGroupNumber|
LearningItemNumber|Priority
MERGE|GlobalAccessGroupRelation|2019/06/13|Rel 1433|OLC143231|OLC901136|2
```

Guidelines for Loading a Community Relation to Oracle Learning

You can link a learning community to one or more course, offering, or specialization items in the Oracle Learning catalog using the CommunityRelation object. You use the community number and learning item numbers to do the linking.

To verify that the relation successfully loaded, use the Learning Communities task on the **My Client Groups** > **Learning** page. You should see the relevant specializations, courses, and communities in the community catalog.

Validation

The load process makes sure that the RelationNumber value doesn't start with OLC.

46 Loading Noncatalog and Legacy Learning Learning Items

Guidelines for Loading a Noncatalog Learning Item to Oracle Learning

You can load learning taken outside of Oracle Learning that needs to appear in the people's learning history using the NoncatalogLearningItem object. You attach the loaded noncatalog learning to learners using the LearningRecord object.

To verify that the noncatalog learning successfully loaded, check for it in the appropriate learner's learning history.

Related Topics

- Guidelines for Loading an Assignment or Recommendation to Oracle Learning

Guidelines for Loading Noncatalog Learning Item Translations to Oracle Learning

You can load translations of noncatalog learning to the other languages installed for Oracle Learning using the NoncatalogLearningItemTranslation object. You need to load the noncatalog learning item before you load the corresponding translations.

To verify that the translations successfully loaded, change the language for Oracle Learning. Then check for the noncatalog learning in the appropriate learner's learning history.

Guidelines for Loading Legacy Learning Items to Oracle Learning

You can load external, completed learning that won't get used going forward, but need to be part of people's learning history using the LegacyLearningItems object. You attach the loaded legacy learning to learners using the LearningRecord object.

The learning won't have any rich media content linked to them because they aren't active items in the Oracle Learning catalog.

To verify that the legacy learning successfully loaded, check for it in the appropriate learner's learning history.

Validations

The load process does these validations:

- The LearningItemNumber attribute has a unique alphanumeric value that doesn't start with the characters OLC.
- The EffectiveStartDate date must be on or before the start date of any effective-dated record that references the legacy learning item record, such as a learning assignment. By default, the effective end date is the end of time. You can't change the effective start and end dates of an existing legacy learning item.
- The OwnedByPersonId value can't be null.
- The Person object exists for the provided PersonId or OwnedByPersonId value.

Related Topics

- Guidelines for Loading an Assignment or Recommendation to Oracle Learning

Example of Loading Legacy Learning Items to Oracle Learning

Here's how you can create a legacy learning item that appears in the applicable learner's learning history. This example identifies the legacy learning item by its source key.

```
COMMENT: Data for Business Object LegacyLearningItem
METADATA|LegacyLearningItem|EffectiveStartDate|LearningItemNumber|Title|ShortDescription|
OwnedByPersonNumber|SourceType|SourceId|SourceInfo|SourceSystemOwner|SourceSystemId
MERGE|LegacyLearningItem|2017/07/26|LEGACY-201710161121-HDL|Title:Legacy LI 201710161121|
DescShort:LEGACY-201710161121|8153756|FUSION_HCM_LEARNING|201710161121|HCM Learn 201710161121 Source Info|
VISION|201710161121
```

Guidelines for Loading Legacy Learning Item Translations to Oracle Learning

You can load translations of legacy learning to the other languages installed for Oracle Learning using the LegacyLearningItemTranslation object. You need to load the legacy learning item before you load the corresponding translations.

To verify that the translations successfully loaded, change the language for Oracle Learning. Then check for the legacy learning in the appropriate learner's learning history.

Example of Loading Legacy Learning Item Translations to Oracle Learning

Here's how you can load translations of the name and description for an existing legacy learning item. This example references the item by its source key.

```
COMMENT: Data for Business Object LegacyLearningItemTranslation
METADATA|LegacyLearningItemTranslation|EffectiveStartDate|Language|LearningItemNumber|Title|
ShortDescription|SourceSystemOwner|SourceSystemId
MERGE|LegacyLearningItemTranslation|2017/07/26|FR|LEGACY-201710161121-HDL|French Title:Legacy LI
 201710161121|French ShortDesc:LEGACY-201710161121|VISION|201710161121
```

47 Loading Course, Offering, Activity, and Specialization Translations to the Learning Catalog

Guidelines for Loading Course Translations to Oracle Learning

You can load translations of a course to the other languages installed for Oracle Learning using the CourseTranslation object. You need to load the course before you load the corresponding translations.

To verify that the translations successfully loaded, change the language for Oracle Learning. Then check the course using the Courses task on the **My Client Groups** > **Learning** page.

Guidelines for Loading Offering Translations to Oracle Learning

You can load translations of an offering to the other languages installed for Oracle Learning using the OfferingTranslation object. You need to load the offering before you load the corresponding translations.

To verify that the translations successfully loaded, change the language for Oracle Learning. Then check the offering using the Offerings task on the **My Client Groups** > **Learning** page.

Guidelines for Loading Instructor-Led Offering Activity Translations to Oracle Learning

You can load translations of an ILT offering activity to the other languages installed for Oracle Learning using the InstructorLedActivityTranslation object.

You need to load the ILT or blended offering and ILT offering activity before you load the corresponding ILT activity translations.

To verify that the translations successfully loaded, change the language for Oracle Learning. Then check the ILT offering activity using the Offerings task on the **My Client Groups** > **Learning** page. On the offering details page, click the Activities tab.

Guidelines for Loading Self-Paced Offering Activity Translations to Oracle Learning

You can load translations of a self-paced offering activity to the other languages installed for Oracle Learning using the SelfPacedActivityTranslation object.

You need to load the self-paced or blended offering and self-paced offering activity before you load the corresponding ILT activity translations.

To verify that the translations successfully loaded, change the language for Oracle Learning. Then check the self-paced offering activity using the Offerings task on the **My Client Groups** > **Learning** page. On the offering details page, click the Activities tab.

Guidelines for Loading Default Offering Pricing Translations to Oracle Learning

You can load translations of default offering pricing to the other languages installed for Oracle Learning using the CourseOfferingPricingDefaults object.

You need to load the course before you load the corresponding default offering pricing translations.

To verify that the translations successfully loaded, change the language for Oracle Learning. Then check the default offering pricing using the Courses task on the **My Client Groups** > **Learning** page.

Guidelines for Loading Specialization Translations to Oracle Learning

You can load translations of a specialization to the other languages installed for Oracle Learning using the SpecializationTranslation object. You need to load the specialization before you load the corresponding translations.

To verify that the translations successfully loaded, change the language for Oracle Learning. Then check the specialization using the Specializations task on the **My Client Groups** > **Learning** page.

Guidelines for Loading Specialization Section Translations to Oracle Learning

You can load translations of a specialization section to the other languages installed for Oracle Learning using the SpecializationSectionTranslation object. You need to load the specialization before you load the corresponding section translations.

To verify that the translations successfully loaded, change the language for Oracle Learning. Then check the specialization section using the Specializations task on the **My Client Groups** > **Learning** page. On the specialization details page, click the Activities tab.

48 Loading Absences Objects

Guidelines for Loading Absence Entry

Using HCM Data Loader, you can load data for the following components of the Absence Entry business object:

Component	Description
Absence Entry	You can load absences for: • **A time-based schedule**: A time-based work schedule has a fixed work day pattern. For example, you define an 8-hour schedule, 5 days a week. You can create a time-based work schedule that starts at 8:00 a.m. and ends at 5:00 p.m. A worker assigned to a time-based work schedule is considered to be available for a fixed number of hours each day. • **An elapsed schedule**: In an elapsed work schedule, workers don't have a fixed start or end time. For example, some workers may start work at 9.00 a.m., and some at 11.00 a.m. A worker assigned to an elapsed work schedule is considered to be available for a number of hours in a day. The **StartDateDuration** and **EndDateDuration** values must be provided. StartDateDuration and EndDateDuration denote duration for a day on Start Date and End Date, respectively. For calendar days UOM, there are 2 options: half day and full day. If you want to record full day absence, StartDateDuration should be given 1 or if absence to be recorded for half day, it should be given 0.5.
Person Childbirth or Placement Absence Record	You can load absence records associated with the birth or placement of a child or children in the worker's household. This record captures planned and actual absence dates, as well as dates associated with the event, such as expected date of childbirth.
Advanced Absence Entry	You can load advanced absence entries when there is a breakdown of absences into specific dates. This is used when an absence is taken against different assignments or if the absence is intermittent. For example, a weekly recurring absence scheduled for a weekly visit to the doctor.

Preparing to Load Absence Entry Objects

Before you load absence records:

- Ensure you provide the name of the employer to verify the employment is valid as of the start date of the absence. The employee's start date determines validation of the absence entry for the legal employer.
- Ensure that the absence type is valid as of the start date of the absence and belongs to the same legislative data group as the person the absence is for.
- Ensure that the absence reason is defined, if it's specified as a required attribute in the absence type.
- Ensure that a work schedule has been assigned to the employee. If there is no work schedule assigned, HDL will take default hours time work schedule.

Deleting Absence Entry Records

You can also delete the Absence Entry business object using HCM Data Loader. When you delete an Absence Entry record, the child records are also deleted.

Example of Loading Accrual Plan Enrollments

You can load absence data for plan enrollment details, such as new enrollments or updates to existing enrollments. Use the Accrual Plan Enrollment business object to:

- Enroll all employees of an organization into an accrual plan in bulk, instead of individually.
- Update employee details when there is a change in their work schedule. You can terminate their enrollment for the previous plan and enroll them into new plans according to the changed work schedule.
- Delete an employee's plan enrollment if there is no accrual record for the corresponding plan.

Preparing to Load Accrual Plan Enrollment Object

Before you load accrual plan enrollment, you must:

- Verify that there are no overlapping enrollment records for the same assignment and plan
- Ensure that the start date is within the employment dates for the employer
- Ensure that both the unit of measure and number of units are specified
- Verify that the plan type is Accrual. This loader does not support other plan types

Loading an Accrual Plan Enrollment

This example PersonAccrualPlanEnrollment.dat file uses user keys to enroll an employee into an accrual plan.

```
METADATA|PersonAccrualPlanEnrollment|PersonNumber|WorkTermAsgNum|PlanName|ManualEnrollment|RunAccruals|
EnrtEndDt|EnrtStDt
MERGE|PersonAccrualPlanEnrollment|955160008177403|ET955160008177403|JL_PLAN_UOM_HRS|N|Y|2018/01/01|
2017/01/01
```

Example of Loading Absence Qualification Plan Entitlement

Using HCM Data Loader, you can load absence entitlement for qualification plans using the Absence Qualification Plan Entitlement business object:

Preparing to Load Absence Qualification Plan Entitlement

Before you load accrual balance details, you must:

- Ensure that the start date and end date are provided because the start and end dates determine entitlement calculation.

- Ensure that the value is negative when loading used entitlements.

Loading an Absence Qualification Plan Entitlement

This example PersonEntitlementDetail.dat file uses source keys to load employee used entitlement details of 5 hours for a sick leave qualification plan.

```
METADATA|PersonEntitlementDetail|SourceSystemOwner|SourceSystemId|PersonId(SourceSystemId)|PlanName|
WorkTermsId(SourceSystemId)|ProcdDate|StartDate|EndDate|PayFactor|Value
MERGE|PersonEntitlementDetail|VISION|ABS02220160411|3423322|SICK_LEAVE|3423322WT|2016/04/11|2016/04/11|
2016/12/31|100|-5
```

Examples of Loading Absence Entry

These example AbsenceEntry.dat files show how to load absence entries using HCM Data Loader for:

- A time based absence entry
- An elapsed absence entry
- A maternity absence entry
- An advanced absence entry

Loading Absence Entry

Example 1: Time Based Absence Entry

This example uses user keys to create an absence entry for two days sick leave.

```
METADATA|PersonAbsenceEntry|Employer|PersonNumber|AbsenceType|AbsenceStatus|ApprovalStatus|StartDate|
StartTime|EndDate|EndTime|Comments|AbsenceReason
MERGE|PersonAbsenceEntry|Vision Corporation|300100005197277|SICK_LEAVE|SUBMITTED|APPROVED|2018/09/24|8:30|
2018/09/25|17:00|SICK LEAVE|SICK_REASON
```

Example 2: Elapsed Absence Entry

This example uses user keys to create an absence entry for two days sick leave.

```
METADATA|PersonAbsenceEntry|PersonNumber|AbsenceType|AbsenceStatus|ApprovalStatus|StartDate|StartTime|
EndDate|EndTime|Comments|StartDateDuration|EndDateDuration|AbsenceReason
MERGE|PersonAbsenceEntry|40010|10026712115|BAT_ABS_TL_Sick|SUBMITTED|APPROVED|2018/01/22||2018/01/23||Sick
 Comments|8|8
```

Loading Person Maternity Absence Entry

This example uses source keys to create an absence entry for maternity leave.

```
METADATA|PersonAbsenceEntry|SourceSystemOwner|SourceSystemId|Employer|PersonId(SourceSystemId)|AbsenceType|
AbsenceStatus|ApprovalStatus|StartDate|StartTime|EndDate|EndTime
MERGE|PersonAbsenceEntry|VISION|AbsMaternity_HDL022|Vision Corporation|HDL022|Maternity Type|SUBMITTED|
APPROVED|2018/01/26|08:00|2018/02/14|17:00
METADATA|PersonMaternityAbsenceEntry|SourceSystemOwner|SourceSystemId|PerAbsenceEntryId(SourceSystemId)|
PlannedStartDate|PlannedReturnDate|LeaveDuration|IntendToWork|ExpectedDateOfChildBirth|ExpectedEndDate|
ActualStartDate|ActualReturnDate|ActualDuration|ActualChildBirthDate|OpenEndedFlag
MERGE|PersonMaternityAbsenceEntry|VISION|AbsMaternityDet_HDL022|AbsMaternity_HDL022|2018/01/26|2018/02/14|||
2018/01/26|2018/02/14|2018/01/25|2018/02/14||2018/01/27|
```

Loading Person Absence Entry Detail

This example uses source keys to create an advanced absence entry.

```
METADATA|PersonAbsenceEntry|Employer|PersonNumber|AbsenceType|AbsenceStatus|ApprovalStatus|StartDate|
StartTime|EndDate|EndTime|Comments|SourceSystemId|SourceSystemOwner|AbsenceReason
MERGE|PersonAbsenceEntry|Vision Corporation|955160008192206|#SK_QUAL_HRS|SUBMITTED|APPROVED|2018/11/06|
21:00|2018/11/07|05:00|Adv night shift|1SCH_SEMP_001111121|VISION|
METADATA|PersonAbsenceEntryDetail|Employer|PersonNumber|AbsenceType|PerAbsEntryDtlId|
PerAbsenceEntryId(SourceSystemId)|AbsenceDate|StartTime|EndDate|EndTime|RowSeq|AbsenceStartDate|
AbsenceStartTime|SourceSystemId|SourceSystemOwner
MERGE|PersonAbsenceEntryDetail|Vision Corporation|955160008192206|#SK_QUAL_HRS||1SCH_SEMP_001111121|
2018/11/06|21:00|||1|2018/11/06|21:00|HDLERDTL05011121|VISION
```

Examples of Loading Carryover Balance

Using HCM Data Loader, you can load absence data for carryover balance for multiple year carryover using the Absence Accrual Balance Component business object. When multi carryover is enabled then HDL load populates in accrual entry details table and anc_per_covr_sub_bal table. Similarly,the application loads ORA_ANC_PREVCOVR in accrual entry details table and anc_per_covr_sub_bal table.

Preparing to Load Carryover Balance

Before you load carryover balance, you must ensure the following:

- Planterm needs to be less than procdate. Planterm and procdate difference shouldn't be more than expiration rule setup

  Plantem is procdDate year - (covr expiration+1). For example, let's assume that procddate is 1/1/2021 and expiration is 2 years setup. The calculation of planterm in this case is: year of procddate (2021) – 2 = 2019
- ProcdDate must be on enrolment start date.
- ORA_ANC_PREVCOVR is a required field.

Loading a Carryover Balance

Example 1

This example uses COVR and PREVCOVR to load carryover balance.

```
METADATA|PersonAccrualDetail|PersonNumber|WorkTermsNumber|PlanName|PlId|AccrualType|ProcdDate|Value|
ElectedAmt|ElectionDt|PerAccrualEntryDtlId|GUID|SourceSystemId|SourceSystemOwner|Planterm
MERGE|PersonAccrualDetail|300100010469809|ET300100010469809|MULTICARRYOVERPLAN||COVR|2021/01/01|100||||||
MERGE|PersonAccrualDetail|300100010469809|ET300100010469809|MULTICARRYOVERPLAN||ORA_ANC_PREVCOVR|2021/01/01|
50||||||2019
MERGE|PersonAccrualDetail|300100010469809|ET300100010469809|MULTICARRYOVERPLAN||ORA_ANC_PREVCOVR|2021/01/01|
50||||||2018
```

Example 2

This example uses user keys to update the existing carryover balance.

```
METADATA|PersonAccrualDetail|PersonNumber|WorkTermsNumber|PlanName|PlId|AccrualType|ProcdDate|Value|
ElectedAmt|ElectionDt|PerAccrualEntryDtlId|GUID|SourceSystemId|SourceSystemOwner|Planterm
MERGE|PersonAccrualDetail|300100010469809|ET300100010469809|MULTICARRYOVERPLAN||COVR|2021/01/01|75||||||
```

```
MERGE|PersonAccrualDetail|300100010469809|ET300100010469809|MULTICARRYOVERPLAN||ORA_ANC_PREVCOVR|2021/01/01|
55||||||2019
MERGE|PersonAccrualDetail|300100010469809|ET300100010469809|MULTICARRYOVERPLAN||ORA_ANC_PREVCOVR|2021/01/01|
55||||||2018
```

49 Loading Benefits Objects

Example of Loading Beneficiary Enrollments

Use the Beneficiary Enrollment business object to designate beneficiaries of multiple employees into a benefit offering.

You typically designate beneficiaries for benefit plans that have payouts in the event of a death or an accident. Similar to participant enrollment, you can load beneficiary data in bulk when you are migrating from a legacy system during a merger or acquisition.

Considerations

You must consider the following points before you load beneficiary enrollment.

- You can load the Beneficiary Enrollment object using either the default benefits relationship or the unrestricted benefits relationship. However, you cannot use both relationships simultaneously.
- You can create and update records using Beneficiary Enrollment HCM Data Loader. You can't delete records.

Preparing to Load Beneficiary Enrollment Object

Before you can load beneficiary enrollment, you must:

- Verify that the data required for enrollment such as employee name, address, salary, and contact details are available in the HR application. It might be difficult to verify errors in HR data after the upload.
- Ensure that you do not include certifications when you are migrating data from a legacy system. This might cause completed enrollments to be suspended again. Avoiding certifications also improves performance and reduces pending action items.
- Determine the number of unique employees and the total number of enrollments that you plan to load into the application. You can obtain this information from the extracted source data.
- Ensure that the total payout percentage assigned to the beneficiaries add up to 100%. Both primary and contingent beneficiaries must have 100% assigned to them.
- Create electable choices on the basis of eligibility rules. To verify whether your electable choices follow eligibility rules, use random checks based on eligibility criteria.
- Create benefits relationships for all the employees whose data you want to load.
- Ensure that the benefit relationships are properly aligned with the hire and assignment dates. For instance, the date on which the life event is processed must follow benefit relationship date, and the benefit relationship date must be the same as or follow the hire date.
- Verify that the total number of potential life events is equal to the total number of employees whose data you want to upload. In the Evaluation and Reporting work area, Processes tab, use the Assign Corrective Life Event process to create the data related to potential life events.
- Run the participation evaluation process without applying the defaults and ensure that the number of started events match with the number of employees you determined in the previous task. You must fix any errors you encounter at this stage.
- Ensure that values are not added to the CloseLifeEvent attributes in the .dat file. You must first verify the enrollment data and then use the **Close Life Event** process to close a life event.

Generating the .dat File

This example BeneficiaryEnrollment.dat file designates a beneficiary into a benefits program.

```
METADATA|BeneficiaryEnrollment|PersonNumber|BenefitRelationship|EffectiveDate|LifeEvent|LifeEventOccuredDate
MERGE|BeneficiaryEnrollment|300100010469441|Default|2015/01/01|PAN_DESG_LE|2015/01/01
METADATA|DesignateBeneficiary|Plan|Program|Option|BeneficiaryPercentage|BeneficiaryPersonNumber|
BeneficiaryType|LineNumber|PersonNumber
MERGE|DesignateBeneficiary|BASIC_LIFE_PLAN|VISION_BENEFITS_PROGRAM|1xSALARY|100|300100010472847|Primary|1|
300100010469441
```

Example of Loading Person Beneficiary Organizations

You can use the Person Beneficiary Organization business object to load details of organizations or trusts in bulk. You can use this object to upload new organizations and trusts or modify existing ones.

Considerations for Using the .dat File

Before you use the .dat file, consider these points:

- In the .dat file, BnfTypCd stands for beneficiary type code. For trusts, use TTEE as the value for BnfTypCd. For organizations, use ORG as the value for BnfTypCd.
- Each row in the .dat file can contain either a beneficiary organization or a trust that you want to link with a participant.
- You can link the same beneficiary organization to more than one participant. To do this, just create a new row in the .dat file with the same details for another participant. Trusts are usually unique for each participant.
- Use PersonNumber to identify the participant.

Examples of Loading Trusts and Organizations

You can use the PersonBeneficiaryOrganization.dat file to load details of trusts and organizations for participants. Let's look at some sample .dat files for loading trusts and organizations.

This example PersonBeneficiaryOrganization.dat file loads details of a trust for the participant:

```
METADATA|PersonBeneficiaryOrganization|StartDate|EndDate|BnfTypCd|TrusteeOrgName|TrusteeOrgRegCd|
TrusteeOrgDescription|TrusteeAddlDetails|TrusteeExecutorName|OrganizationName|PersonNumber

MERGE|PersonBeneficiaryOrganization|2018/05/05||TTEE|This is Duprey Trust|TOR111|TOD|TAD|TEN||10026712115
```

This example PersonBeneficiaryOrganization.dat file loads details of an organization for the participant:

```
METADATA|PersonBeneficiaryOrganization|StartDate|EndDate|BnfTypCd|TrusteeOrgName|TrusteeOrgRegCd|
TrusteeOrgDescription|TrusteeAddlDetails|TrusteeExecutorName|OrganizationName|PersonNumber

MERGE|PersonBeneficiaryOrganization|2019/07/07||ORG|||||UnitedWay|10026712115
```

Example of Loading Dependent Enrollments

Use the Dependent Enrollment business object to designate dependents of multiple employees into a benefit offering.

You typically perform this task when you are migrating from a legacy system during a merger or acquisition. You can also use the data loader if you want to move dependents in bulk from one plan to another.

Considerations

You must consider the following points before you load dependent enrollment.

- You can load the Dependent Enrollment object using either the default benefits relationship or the unrestricted benefits relationship. However, you cannot use both relationships simultaneously.
- You can create and update records using Dependent Enrollment HCM Data Loader. You can't delete records.
- Dependent Enrollment processing does not depend on Effective date, but on Life Event date.

Preparing to Load Dependent Enrollment Object

Before you can load dependent enrollment, you must:

- Verify that the data required for enrollment such as employee name, address, salary, and contact details are available in the HR application. It might be difficult to verify errors in HR data after the upload.
- Ensure that you do not include certifications when you are migrating data from a legacy system. This might cause completed enrollments to be suspended again. Avoiding certifications also improves performance and reduces pending action items.
- Determine the number of unique employees and the total number of enrollments that you plan to load into the application. You can obtain this information from the extracted source data.
- Ensure that the total payout percentage assigned to the beneficiaries add up to 100%. Both primary and contingent beneficiaries must have 100% assigned to them.
- Create electable choices on the basis of eligibility rules. To verify whether your electable choices follow eligibility rules, use random checks based on eligibility criteria.
- Create benefits relationships for all the employees whose data you want to load.
- Ensure that the benefit relationships are properly aligned with the hire and assignment dates. For instance, the date on which the life event is processed must follow benefit relationship date, and the benefit relationship date must be the same as or follow the hire date.
- Verify that the total number of potential life events is equal to the total number of employees whose data you want to upload. In the Evaluation and Reporting work area, Processes tab, use the Assign Corrective Life Event process to create the data related to potential life events.
- Run the participation evaluation process without applying the defaults and ensure that the number of started events match with the number of employees you determined in the previous task. You must fix any errors you encounter at this stage.
- Ensure that values are not added to the CloseLifeEvent attributes in the .dat file. You must first verify the enrollment data and then use the **Close Life Event** process to close a life event.

Generate the .dat File

This example DependentEnrollment.dat file designates a dependent into a benefits program.

```
METADATA|DependentEnrollment|PersonNumber|BenefitRelationship|EffectiveDate|LifeEvent|LifeEventOccuredDate
MERGE|DependentEnrollment|10026712115|Default|2016/10/28|NewHire|2016/10/28
METADATA|DesignateDependent|Plan|Program|Option|DependentPersonNumber|LineNumber|PersonNumber
MERGE|DesignateDependent|GlobalPlan|Green Company Full Benefits|Employee plus Family|100010026713030|1|
10026712115
```

Example of Loading Participant Enrollments

Use the Participant Enrollment business object to enroll participants into a benefit offering in bulk.

Considerations

You must consider the following points before you load participant enrollment.

- You can load the Participant Enrollment object using either the default benefits relationship or the unrestricted benefits relationship. However, you cannot use both relationships simultaneously.
- You can create records using Participant Enrollment HCM Data Loader. You can't update or delete records.
- In order to update a record, de-enroll the participant first and then re-enroll the participant with the new value. Refer to the example ParticipantEnrollment.dat files provided in this topic to de-enroll and re-enroll a participant.

Preparing to Load Participant Enrollment Object

Before you can load participant enrollment, you must:

- Verify that the data required for enrollment such as employee name, address, salary, and contact details are available in the HR application. It might be difficult to verify errors in HR data after the upload.
- Ensure that you do not include certifications when you are migrating data from a legacy system. This might cause completed enrollments to be suspended again. Avoiding certifications also improves performance and reduces pending action items.
- Determine the number of unique employees and the total number of enrollments that you plan to load into the application. You can obtain this information from the extracted source data.
- Ensure that the total payout percentage assigned to the beneficiaries add up to 100%. Both primary and contingent beneficiaries must have 100% assigned to them.
- Create electable choices on the basis of eligibility rules. To verify whether your electable choices follow eligibility rules, use random checks based on eligibility criteria.
- Create benefits relationships for all the employees whose data you want to load.
- Ensure that the benefit relationships are properly aligned with the hire and assignment dates. For instance, the date on which the life event is processed must follow benefit relationship date, and the benefit relationship date must be the same as or follow the hire date.
- Verify that the total number of potential life events is equal to the total number of employees whose data you want to upload. In the Evaluation and Reporting work area, Processes tab, use the Assign Corrective Life Event process to create the data related to potential life events.

- Run the participation evaluation process without applying the defaults and ensure that the number of started events match with the number of employees you determined in the previous task. You must fix any errors you encounter at this stage.
- Ensure that values are not added to the CloseLifeEvent attributes in the .dat file. You must first verify the enrollment data and then use the **Close Life Event** process to close a life event.

Generating the .dat File

This example ParticipantEnrollment.dat file enrolls a participant into a benefits program.

```
METADATA|ParticipantEnrollment|PersonNumber|ParticipantLastName|ParticipantFirstName|BenefitRelationship|
LifeEvent|LifeEventOccuredDate|EffectiveDate
MERGE|ParticipantEnrollment|10026712115|Duprey|Luis|Default|NewHire|2016/10/28|2016/11/05
METADATA|CompensationObject|Program|OriginalEnrollmentDate|PersonNumber|LineNumber
MERGE|CompensationObject|Green Company Full Benefits|2014/10/28|10026712115|1
```

Note: It is not necessary to supply the ParticipantLastName and ParticipantFirstName attributes. The application does not perform any validation to cross check the name against the person number. These attributes are for provided only for your information.

De-enrolling from an Existing Option in the Plan and Enrolling in a Waive Option

You can de-enroll employees from a plan and enroll them into the waive option of same plan type. For instance, consider an employee who is enrolled in the Healthy HMO plan with the Employee Plus Family option. You can de-enroll the employee from the Employee Plus Family option and enroll the employee in the Waive Coverage option using the following example ParticipantEnrollment.dat file.

```
METADATA|ParticipantEnrollment|PersonNumber|ParticipantLastName|ParticipantFirstName|BenefitRelationship|
LifeEvent|LifeEventOccuredDate|EffectiveDate
MERGE|ParticipantEnrollment|10026712115|Duprey|Luis|Default|Gain Dependent|2016/10/28|2016/10/28
METADATA|CompensationObject|Plan|Program|Option|OriginalEnrollmentDate|PersonNumber|DenrollPlan|
DenrollOption|LineNumber
MERGE|CompensationObject|Healthy HMO|Healthy Benefits|Waive Coverage|2016/10/28|10026712115|Healthy HMO|
Employee plus Family|1
```

Example of Loading Person Benefit Group

You create a benefit group to determine a specific set of people who are eligible for a benefit object. Use the Person Benefit Group business object to load multiple benefits groups simultaneously using HCM Data Loader.

Preparing to Load Person Benefit Group Object

Verify that the data required for loading a business group, such as the person name, is available in the HR application. It might be difficult to verify errors in HR data after the upload.

Generating the .dat File

This example PersonBenefitGroup.dat file loads multiple benefit groups.

```
METADATA|PersonBenefitGroup|SourceSystemOwner|SourceSystemId|BenefitGroupName|EffectiveStartDate|PersonId
```

```
MERGE|PersonBenefitGroup|VISION|GROUP955160008173169|VISION_BENEFIT_GROUP|2019/01/01|955160008173160
MERGE|PersonBenefitGroup|VISION|GROUP100000012556816|VISION_BENEFIT_GROUP|2019/01/01|100000012556816
```

Example of Loading Person Benefit Balance

Use the Person Benefit Balance business object to load multiple benefits balances using HCM Data Loader. You typically use benefits balances to store benefit amounts from an external system, or as a workaround for a calculation to bridge disparate systems.

Preparing to Load Person Benefit Balance Object

Verify that the data required for uploading benefit balances such as benefit balance, legal entity, and assignment details are available in the HR application. It might be difficult to verify errors in HR data after the upload.

Generate the .DAT File

This example PersonBenefitBalance.dat file loads a person benefit balance.

```
METADATA|PersonBenefitBalance|SourceSystemOwner|SourceSystemId|PersonId|BenefitBalanceName|
BenefitRelationName|EffectiveStartDate|UOM|Val|
MERGE|PersonBenefitBalance|VISION|BALANACE955160008173169|955160008173169|SICK_LEAVE_BALANCE|DFLT|
2009/01/01|USD|2200
```

Example of Loading Person Habits

Use the Person Habits business object to load details of habits and disability statuses of employees in bulk using HCM Data Loader.

Preparing to Load Person Habits

Before you can load person habits, you must:

- Verify that the objects referenced by Person Habits exists in the target environment. If they do not exist, the load will fail.
- Create electable choices on the basis of eligibility rules. Lack of eligibility causes a data roll back for corresponding employees. To verify whether your electable choices follow eligibility rules, use random checks based on eligibility criteria.

Generating the .dat File

This example PersonHabits.dat file loads the person habits for an employee who is a full time student, smokes and is registered as disabled.

```
METADATA|PersonHabits|SourceSystemOwner|SourceSystemId|PersonNumber|EffectiveStartDate|LegalEmployer|
DisabilityStatus|RegisteredDisabledFlag|StudentStatus|TobaccoTypeUsage
MERGE|PersonHabits|VISION|HABIT955160008173169|955160008173169|2015/01/01|Vision_Corporation|A|Y|FULL_TIME|Y
```

50 Loading Recruiting Objects

Candidates

Overview of Loading Candidates

You can load candidate details such as: Candidate Name, Candidate Email, Candidate Address, Candidate Phone, Candidate Extra Information, Candidate Interaction, Candidate National Identifier, Candidate Profile, Attachment, and, Candidate Preference such as the Candidate Preferred Job Family, Candidate Preferred Location and Candidate Preferred Organization using the HCM Data Loader (HDL).

The Candidate object is a parent object with multiple child objects. To view this structure:

1. Go to **My Client Groups** > **Data Exchange** > **View Business Objects**.
2. Type **Recruiting** in the text box above Product Area and **Candidate** in the text box above Business Object, and press **Enter**.
3. From the search results, click the **Candidate** business object. The structure of the Candidate object with its child objects appears in the Components pane.
4. Click each child object to view its component details and attributes.

Guidelines for Loading Candidates

This topic describes the considerations for loading Candidates using HCM Data Loader (HDL).

Candidate Number

You should provide the candidate number as a combination of alphanumeric and special characters underscore (_) or hyphen (-).

Candidate Confirmed indicator and Start Date

You should set the candidate confirmed indicator to 'Y' and ensure that the start date of the candidate isn't greater than the current date for the candidate to be searchable.

Deleting External Candidate

You can delete an external candidate only if there are no confirmed active job applications for the candidate.

Setting the Preferred Mode of Communication

You can either set the email address or the phone number of the candidate as a preferred mode of communication. If the preferred mode isn't specified in the candidate.dat file, then the email address of the candidate is chosen by default. Duplicate checks are run on the preferred mode.

Note: You can only load external candidates using candidate.dat. Internal candidates are loaded using worker.dat. Phone numbers uploaded via HCM Data Loader are considered as verified and correct. Hence, candidates won't be able to edit them on external career sites. If you want the phone number to be editable, don't load it via HCM Data Loader.

AddToPoolName Attribute

You should remove this attribute while loading the Candidate.dat file for a specific candidate for the second time. This attribute is provided as a quick measure to add the candidate to a pool and therefore should not be reused on the same candidate in the second HDL load.

Related Topics

- Overview of Loading Workers

Examples of Loading Candidates

This topic provides examples that show how to load candidates.

Creating Candidates

This example Candidate.dat file creates a basic candidate record with the attributes CandidateNumber, CandidateEmail, CandidateName, CandidatePhone and CandidateAddress.

```
METADATA|Candidate|CandidateNumber|PrefPhoneCntTypeCode|ObjectStatus|AvailabilityDate|StartDate|
CountryOfBirth|RegionOfBirth|TownOfBirth|AddedByFlowCode|ConfirmedFlag|VisibleToCandidateFlag|
CandPrefLanguageCode|SourceSystemId|SourceSystemOwner
MERGE|Candidate|CAN001|HM|Active|2017/11/01 00:00:00|2015/12/01|US|Colorado|Denver|ORA_INTERNAL|Y|Y|US|
CAN001|HRC_SQLLOADER
METADATA|CandidateEmail|SourceSystemId|DateFrom|EmailAddress|EmailType|SourceSystemId|SourceSystemOwner
MERGE|CandidateEmail|CAN001|2015/12/01|CAN001@invalidemail.com|Home Email|CAN001_EMAIL|HRC_SQLLOADER
METADATA|CandidateName|EffectiveStartDate|CandidateNumber|NameType|FirstName|LastName|SourceSystemId|
SourceSystemOwner
MERGE|CandidateName|2015/12/01|CAN001|GLOBAL|CAN001_FN|CAN001_LN|CAN001_NAME|HRC_SQLLOADER
METADATA|CandidatePhone|CandidateNumber|AreaCode|PhoneNumber|PhoneType|DateFrom|SourceSystemId|
SourceSystemOwner
MERGE|CandidatePhone|CAN001|749|3111227|HM|2015/12/01|CAN001_PHONE|HRC_SQLLOADER
METADATA|CandidateAddress|CandidateNumber|AddressLine1|AddressLine2|AddressLine3|AddressLine4|Building|
FloorNumber|TownOrCity|Region1|Region2|Region3|Country|PostalCode|EffectiveStartDate|AddressType|
SourceSystemId|SourceSystemOwner
MERGE|CandidateAddress|CAN001|879 Fyxaiba Street|Suite 4242|||||FLORENCE|BOONE|KY||US|41022|2015/12/01|HOME|
CAN001_ADDR|HRC_SQLLOADER
```

This example Candidate.dat file creates a comprehensive candidate record with all its child attributes CandidateNumber, CandidateEmail, CandidateName, CandidatePhone, CandidateAddress, CandidateProfile, CandidateInteraction, CandidatePreference, CandidatePreferredJobFamily, CandidatePreferredLocation, and Attachment.

```
METADATA|Candidate|CandidateNumber|PrefPhoneCntTypeCode|ObjectStatus|AvailabilityDate|StartDate|
CountryOfBirth|RegionOfBirth|TownOfBirth|AddedByFlowCode|ConfirmedFlag|VisibleToCandidateFlag|
CandPrefLanguageCode|SourceSystemId|SourceSystemOwner
MERGE|Candidate|CAN002|HM|Active|2017/11/01 00:00:00|2015/12/01|US|Colorado|Denver|ORA_INTERNAL|Y|Y|US|
CAN002|HRC_SQLLOADER
METADATA|CandidateEmail|SourceSystemId|DateFrom|EmailAddress|EmailType|SourceSystemId|SourceSystemOwner
MERGE|CandidateEmail|CAN002|2015/12/01|CAN002@invalidemail.com|Home Email|CAN002_EMAIL|HRC_SQLLOADER
METADATA|CandidateName|EffectiveStartDate|CandidateNumber|NameType|FirstName|LastName|SourceSystemId|
SourceSystemOwner
MERGE|CandidateName|2015/12/01|CAN002|GLOBAL|CAN002_FN|CAN002_LN|CAN002_NAME|HRC_SQLLOADER
```

```
METADATA|CandidatePhone|CandidateNumber|AreaCode|PhoneNumber|PhoneType|DateFrom|SourceSystemId|
SourceSystemOwner
MERGE|CandidatePhone|CAN002|749|3111228|HM|2015/12/01|CAN002_PHONE|HRC_SQLLOADER
METADATA|CandidateAddress|CandidateNumber|AddressLine1|AddressLine2|AddressLine3|AddressLine4|Building|
FloorNumber|TownOrCity|Region1|Region2|Region3|Country|PostalCode|EffectiveStartDate|AddressType|
SourceSystemId|SourceSystemOwner
MERGE|CandidateAddress|CAN002|879 Fyxaiba Street|Suite 4242|||||FLORENCE|BOONE|KY||US|41022|2015/12/01|HOME|
CAN002_ADDR|HRC_SQLLOADER
METADATA|CandidateProfile|CandidateNumber|ContentTypeId|ContentItem|ContentType|CountryId|Importance|
DateFrom|DateTo|ItemDate1|ItemDate2|ItemDate3|ItemDate4|ItemDate6|ItemDate7|ItemDate8|ItemDecimal1|
ItemNumber1|ItemNumber2|ItemNumber3|ItemNumber4|ItemNumber5|ItemNumber8|ItemNumber9|ItemNumber10|
ItemText20001|ItemText20003|ItemText20004|ItemText20005|ItemText2401|ItemText2402|ItemText2403|
ItemText2404|ItemText2405|ItemText2406|ItemText2407|ItemText2408|ItemText2409|ItemText24012|ItemText24013|
ItemText24014|ItemText301|ItemText302|ItemText303|ItemText304|ItemText305|ItemText306|ItemText307|
ItemText308|ItemText309|ItemText3010|ItemText3011|ItemText3012|ItemText3013|RatingLevelId1|RatingLevelId2|
RatingLevelId3|RatingModelId1|RatingModelId2|RatingModelId3|ItemClob1|ItemClob2|ItemClob3|ItemClob4|
ItemClob5|SourceSystemId|SourceSystemOwner
MERGE|CandidateProfile|CAN002|109|Telugu|Languages||2|2018/06/22||2018/05/06|||||||||||||||||||||||||||||||
Y|N|Y|||||||||||8002|8003|8001|8|8|8|Clob1.txt|Clob2.txt|Clob3.txt|Clob4.txt|Clob5.txt|CAN002_LANG_0|
HRC_SQLLOADER
MERGE|CandidateProfile|CAN002|117||||2|2018/06/21|||||||||100000011915673|100000012474304|100000015162235|
100000011915004|||130000|77|||||Relocation Reason Text 123|||||||||||Y|75PCT|Y|25PCT|Y||Y|Y|Y|PKR|
M|||||||||Clob1.txt|Clob2.txt|Clob3.txt|Clob4.txt|Clob5.txt|CAN002_WORKEXP_0|HRC_SQLLOADER
MERGE|CandidateProfile|CAN002|106|ML - Masters Level Degree|Degrees|1|2|2018/06/25|||||2015/04/30|
2016/06/08|2013/07/01|2016/05/31|9.98|77777|44444|99999|||2016|300100153827153||School Name Test 123|
Comments Comments Comments||Reimbursement Arrangements Comments|Major Test 123||Awarding Body Test 123|
Educator Test 123|Minor Test 123|A|55555||Faculty or Department Test 123|City Test 123|||Y|N|SAR|GOOD|N||
Class||||Active|Y|I|||||||Clob1.txt|Clob2.txt|Clob3.txt|Clob4.txt|Clob5.txt|CAN002_DEGREE_0|HRC_SQLLOADER
MERGE|CandidateProfile|CAN002|129||||2|2018/06/25||2013/07/13|2015/07/12|||||||||||300100148485144|||
Gained Knowledge on C++||||Previous Employer Name Test 1|Other Job Function Test 1|7777777777|
SupervisorEmailTest1@oracle.com||Supervisor Name Test 1|Supervisor Title Test 1|Starting Position Test 1|
Junior QA|Employer City Test 1|Type of Business Test 1|Other Compensation Test 123|N|Y|||||MANAG|First IT
 Services|||||||||||Clob1.txt|Clob2.txt|Clob3.txt|Clob4.txt|Clob5.txt|CAN002_PREVEMP_0|HRC_SQLLOADER
MERGE|CandidateProfile|CAN002|103|Oracle Certified Professional (OCP)|Licenses and Certifications|10099|
2|2018/06/25||||2019/06/20|2020/12/13|2020/07/11|||||1221123|1221140||||300100153573741||Educational
 Establishment Test 123|Comments Comments|Restrictions Comments Values||Cardio Pulmonary Title|Certificate
 Number Test 123||Issued By Test 123|||123||||||Y||||N|Y|||||||||||||Clob1.txt|Clob2.txt|Clob3.txt|
Clob4.txt|Clob5.txt|CAN002_LIC_CERT_0|HRC_SQLLOADER
METADATA|CandidateInteraction|AddedByPersonId|InteractionDate|InteractionId|InteractionTypeCode|PersonId|
CandidateNumber|Text|SourceSystemId|SourceSystemOwner
MERGE|CandidateInteraction|300100148463523|2018/01/01 12:00:00||ORA_EMAIL||CAN002|SentEmail.txt|CAN002-
Interaction1_SysId|HRC_SQLLOADER
METADATA|CandidatePreference|CandidateNumber|OptInTcEmailsDate|OptInTcEmailsFlag|SiteNumber|TcConfirmedFlag|
TcConfirmedDate|SourceSystemId|SourceSystemOwner
MERGE|CandidatePreference|CAN002|2018/10/01 12:30:22|Y|CX|Y|2018/10/10 12:30:30|CAN002-CandPref_SysId|
HRC_SQLLOADER
METADATA|CandidatePreferredJobFamily|CandidateNumber|JobFamilyName|SiteNumber|SourceSystemId|
SourceSystemOwner
MERGE|CandidatePreferredJobFamily|CAN002|ZFRCE_Planning|CX|CAN002-CandPrefJobFam1_SysId|HRC_SQLLOADER
MERGE|CandidatePreferredJobFamily|CAN002|ZFRCE_Supply Chain and Distribution|CX|CAN002-
CandPrefJobFam2_SysId|HRC_SQLLOADER
METADATA|CandidatePreferredLocation|CandidateNumber|LocationName|SiteNumber|SourceSystemId|SourceSystemOwner
MERGE|CandidatePreferredLocation|CAN002|United States|CX|CAN002-CandPrefLoc1_SysId|HRC_SQLLOADER
MERGE|CandidatePreferredLocation|CAN002|Canada|CX|CAN002-CandPrefLoc2_SysId|HRC_SQLLOADER
METADATA|Attachment|File|DataTypeCode|CandidateNumber|Title|URLorFileName|Category
MERGE|Attachment|CAN002Resume.txt|FILE||CAN002|TestAttach.txt|CAN002Resume.txt|IRC_CANDIDATE_RESUME
```

Adding Candidates to Candidate Pool

This example Candidate.dat file creates a candidate record and adds it to a candidate pool using the AddToPoolName and PoolOwnerPersonName attributes.

> **Note:** When you are loading the Candidate.dat file using HDL for the second time for a specific candidate, ensure that you remove the AddToPoolName attribute for that candidate and then reload the file.

```
METADATA|Candidate|CandidateNumber|PrefPhoneCntTypeCode|ObjectStatus|AvailabilityDate|StartDate|
CountryOfBirth|RegionOfBirth|TownOfBirth|AddedByFlowCode|ConfirmedFlag|VisibleToCandidateFlag|
CandPrefLanguageCode|SourceSystemId|SourceSystemOwner|AddToPoolName|PoolOwnerPersonNumber
MERGE|Candidate|CAN003|HM|Active|2017/11/01 00:00:00|2015/12/01|US|Colorado|Denver|ORA_INTERNAL|Y|Y|US|
CAN003|HRC_SQLLOADER|POOL001|POOLOWNER01
METADATA|CandidateName|EffectiveStartDate|CandidateNumber|NameType|FirstName|LastName|SourceSystemId|
SourceSystemOwner
MERGE|CandidateName|2015/12/01|CAN003|GLOBAL|CAN003_FN|CAN003_LN|CAN003_NAME|HRC_SQLLOADER
METADATA|CandidatePhone|CandidateNumber|AreaCode|PhoneNumber|PhoneType|DateFrom|SourceSystemId|
SourceSystemOwner
MERGE|CandidatePhone|CAN003|749|3111229|HM|2015/12/01|CAN003_PHONE|HRC_SQLLOADER
```

Updating Candidates

This example Candidate.dat file updates the CandidateAddress of a candidate record.

```
METADATA|CandidateAddress|CandidateNumber|AddressLine1|AddressLine2|AddressLine3|AddressLine4|Building|
FloorNumber|TownOrCity|Region1|Region2|Region3|Country|PostalCode|EffectiveStartDate|AddressType|
SourceSystemId|SourceSystemOwner
MERGE|CandidateAddress|CAN002|879 Fyxaiba Street|Suite 4253|||||FLORENCE|BOONE|KY||US|41024|2015/12/01|HOME|
CAN002_ADDR|HRC_SQLLOADER
```

Deleting Candidates

This example Candidate.dat file deletes the candidate record using the CandidateNumber attribute.

```
METADATA|Candidate|CandidateNumber
DELETE|Candidate|CAN_TBD-01
```

Candidate Job Applications

Overview of Loading Candidate Job Applications

Using the HCM Data Loader (HDL), you can create, update and delete candidate job applications.

You can add or update all artifacts related to job applications, including responses to interview questions.

Guidelines for Loading Candidate Job Applications

This topic describes the considerations for loading job applications using HCM Data Loader.

Considerations for Creating Candidate Job Applications

- Load only currently active job applications. It's recommended not to load historic job applications for reporting purposes.
- Provide the expected `Source` and `SourceMedium` attributes while loading candidate job applications. If you don't provide these, the `Source` attribute defaults to **internal career site** and `SourceMedium` attribute defaults to **career site**.
- Job applications can be loaded only to the first phase and state of the Candidate Selection Process (CSP).

- While migrating job applications from a legacy system, load to the first phase and state of the CSP and later auto-progress to the target phase and state or manually move using HDL following the CSP rules.

  > **Tip:** Configure a flex field on the job application to flag it as a migrated job application and then build a fast formula to read the migration flag, which auto-progresses to the target phase and state.

- Loading job applications with offers isn't supported. It's recommended to load accepted offers as New Hires or Pending Workers.

User Key

You must provide both these user keys while loading a job application:

- RequisitionNumber
- CandidateNumber

Considerations for Deleting Candidate Job Applications

- A job application can be deleted by using the `InactivateFlag` attribute. This effects only a soft deletion of the job application.
- The job application has to be in a terminal state before you can delete it.

Considerations for Moving Candidate Job Applications

Using HCM Data Loader, you can move job applications from the current phase and state to a target phase and state. However, the move to a target phase is validated and allowed based on the rules set in the CSP. You can't move a job application past a mandatory phase or out of a state without meeting the required condition set in the CSP.

You can move a job application to the Offer phase and to states within the Offer phase based on these set of rules. You can also move a job application from the Offer phase to a custom phase between the Offer and HR phases.

> **Note:** The movement of a job application to a target phase and state is recorded as on the date and time of the load.

Job Application Movements in the Offer Phase

The following moves are supported in the Offer phase:

From State	To State in Offer Phase
Any state in a phase prior to the Offer phase	• To Be Created • Draft
To Be Created	• Rejected (by Employer) • Withdrawn by Candidate • Draft
Draft	• Rejected (by Employer) • Withdrawn by Candidate

From State	To State in Offer Phase
Pending Approval	• Draft
Approval Rejected	• Draft • Rejected (by Employer) • Withdrawn by Candidate
Approved	• Extended, if the Extend Bypass indicator isn't selected **Note:** On moving an offer to the Extended state, the candidate receives the offer and is also notified. • Accepted, if the Extend Bypass indicator is selected • Draft • Rejected (by Employer) • Withdrawn by Candidate
Extended	• Rejected (by Employer) • Withdrawn by Candidate • Accepted
Accepted	• Rejected (by Employer) • Withdrawn by Candidate • Post-offer phase and state: This is any customer-configured post-offer phase.
Rejected (by Employer)	• Draft, if the offer exists • To Be Created, if the offer doesn't exist
Withdrawn by Candidate	• Draft, if the offer exists • To Be Created, if the offer doesn't exist

Job Application Movements in the HR Phase

Once the offer is accepted by the candidate, you can move the offer to the HR phase and Pending Automated Processing state using HCM Data Loader. The offer is then selected for automated processing.

If an error occurs during HR processing, you can move a job application to one of the terminal states, Rejected by Employer or Withdrawn by Candidate, using HCM Data Loader.

Examples of Loading Candidate Job Applications

This topic provides sample files that show how to create and delete job applications, and also move them to specific phases and states that are supported in HCM Data Loader.

Creating Candidate Job Applications

Job Application with Basic Information

This sample CandidateJobApplication.dat file creates a basic job application using the RequisitionNumber, CandidateNumber, SubmissionLanguageCode, SourceSystemId, and SourceSystemOwner attributes.

```
METADATA|CandidateJobApplication|RequisitionNumber|CandidateNumber|SubmissionLanguageCode|SourceSystemId|
SourceSystemOwner

MERGE|CandidateJobApplication|HDLJobRequisition|CAN001|KO|CAN001-HDLJobRequisition|HRC_SQLLOADER
```

Job Application with Full Information

This sample CandidateJobApplication.dat file creates a full job application, including child objects such as Attachment and Candidate Job Application Profile. The sample file contains the required attributes required for the load.

```
METADATA|CandidateJobApplication|RequisitionNumber|CandidateNumber|SubmissionLanguageCode|SourceSystemId|
SourceSystemOwner|Source|SourceMedium
METADATA|Attachment|Title|File|DataTypeCode|URLorFileName|RequisitionNumber|CandidateNumber|Category
METADATA|CandidateJobApplicationProfile|RequisitionNumber|CandidateNumber|ContentTypeId|ContentItem|
ContentType|CountryId|Importance|DateFrom|DateTo|ItemDate1|ItemDate2|ItemDate3|ItemDate4|ItemDate6|
ItemDate7|ItemDate8|ItemDecimal1|ItemNumber1|ItemNumber2|ItemNumber3|ItemNumber4|ItemNumber5|ItemNumber8|
ItemNumber9|ItemNumber10|ItemText20001|ItemText20003|ItemText20004|ItemText20005|ItemText2401|ItemText2402|
ItemText2403|ItemText2404|ItemText2405|ItemText2406|ItemText2407|ItemText2408|ItemText2409|ItemText24012|
ItemText24013|ItemText24014|ItemText301|ItemText302|ItemText303|ItemText304|ItemText305|ItemText306|
ItemText307|ItemText308|ItemText309|ItemText3010|ItemText3011|ItemText3012|ItemText3013|RatingLevelId1|
RatingLevelId2|RatingLevelId3|RatingModelId1|RatingModelId2|RatingModelId3|ItemClob1|ItemClob2|ItemClob3|
ItemClob4|ItemClob5|SourceSystemId|SourceSystemOwner|CountryCountryCode|StateGeographyCode

MERGE|CandidateJobApplication|HDLJobRequisition|CAN001|KO|SUB-ZFRCE-HDLJOBREQ|HRC_SQLLOADER|email|campaign
MERGE|Attachment|CAN001-Resume-File|Resume.rtf|FILE|Resume.rtf|HDLJobRequisition|CAN001|IRC_CANDIDATE_RESUME
MERGE|Attachment|CAN001-Resume-Link||WEB_PAGE|http://www.google.com|HDLJobRequisition|CAN001|
IRC_CANDIDATE_RESUME
MERGE|Attachment|CAN001-CoverLetter-File|Cover-Letter.doc|FILE|Cover-Letter.doc|HDLJobRequisition|CAN001|
IRC_CANDIDATE_COVERLETTER
MERGE|Attachment|CAN001-CoverLetter-Link||WEB_PAGE|http://www.oracle.com|HDLJobRequisition|CAN001|
IRC_CANDIDATE_COVERLETTER
MERGE|Attachment|CAN001-Social-Link||WEB_PAGE|http://www.facebook.com|HDLJobRequisition|CAN001|
IRC_CANDIDATE_SOCIAL
MERGE|Attachment|CAN001-InternalDoc-File|Internal-Document.docx|FILE|Internal-Document.docx|
HDLJobRequisition|CAN001|IRC_INTERNAL
MERGE|Attachment|CAN001-InternalDoc-Link||WEB_PAGE|http://www.yahoo.com|HDLJobRequisition|CAN001|
IRC_INTERNAL
MERGE|CandidateJobApplicationProfile|HDLJobRequisition|CAN001|109|Korean|Languages||2|2018/06/22||
2018/05/06|||||||||||||||||||||||||||||||Y|N|Y|||||||||||8001|8002|8003|8|8|8|Clob1.txt|Clob2.txt|
Clob3.txt|Clob4.txt|Clob5.txt|SUBPROF-LANG-ZFRCE-HDLJOBREQ|HRC_SQLLOADER|US|GU
MERGE|CandidateJobApplicationProfile|HDLJobRequisition|CAN001|117||||2|2018/06/21|||||||||100000011915673|
100000012474304|100000015162235|100000011915004|||130000|77|||||JA Relocation Reason Text 123|||||||||||
Y|25PCT|Y|75PCT|Y||Y|Y|Y|PKR|M|||||||||Clob1.txt|Clob2.txt|Clob3.txt|Clob4.txt|Clob5.txt|SUBPROF-WORKREQ-
ZFRCE-HDLJOBREQ|HRC_SQLLOADER|US|CA
MERGE|CandidateJobApplicationProfile|HDLJobRequisition|CAN001|106|ML - Masters Level Degree|Degrees|1|2|
2018/06/25|||||2015/04/30|2016/06/08|2013/07/01|2016/05/31|9.98|77777|44444|99999|||2016|300100153827153||
JA School Name Test 123|JAComments Comments Comments||JA Reimbursement Arrangements Comments|JA Major Test
 123||JA Awarding Body Test 123|JA Educator Test 123|JA Minor Test 123|A|55555||JA Faculty or Department
 Test 123|JA City Test 123|||Y|N|SAR|GOOD|N||Class||||Active|Y|I|||||||Clob1.txt|Clob2.txt|Clob3.txt|
Clob4.txt|Clob5.txt|SUBPROF-DEGREE-ZFRCE-HDLJOBREQ|HRC_SQLLOADER|US|MA
MERGE|CandidateJobApplicationProfile|HDLJobRequisition|CAN001|129||||2|2018/06/25||2013/07/13|
2015/07/12|||||||||||||300100148485144|||JA Gained Knowledge on C++||||JA Previous Employer Name Test 1|
JA Other Job Function Test 1|7777777777|JASupervisorEmailTest1@oracle.com||JA Supervisor Name Test 1|JA
 Supervisor Title Test 1|JA Starting Position Test 1|Junior QA|JA Employer City Test 1|JA Type of Business
 Test 1|JA Other Compensation Test 123|N|Y|||||MANAG|First IT Services|||||||||||Clob1.txt|Clob2.txt|
Clob3.txt|Clob4.txt|Clob5.txt|SUBPROF-PREVEMP-ZFRCE-HDLJOBREQ|HRC_SQLLOADER|US|IA
```

```
MERGE|CandidateJobApplicationProfile|HDLJobRequisition|CAN001|103|Oracle Certified Professional (OCP)|
Licenses and Certifications|10099|2|2018/06/25||||2019/06/20|2018/12/13|2017/07/11|||||1221123|1221140||||
300100153573741||JA Educational Establishment Test 123|Comments Comments|JA Restrictions Comments
 Values||Cardio Pulmonary Title|JA Certificate Number Test 123||JA Issued By Test 123|||123||||||Y||||N|
Y|||||||||||||||Clob1.txt|Clob2.txt|Clob3.txt|Clob4.txt|Clob5.txt|SUBPROF-CERT-ZFRCE-HDLJOBREQ|HRC_SQLLOADER|
AU|ACT
```

Job Application with Feedback Responses

This sample CandidateJobApplication.dat file shows how to upload feedback responses received along with rating from the hiring team.

```
METADATA|CandidateJobApplication|RequisitionNumber|CandidateNumber|SubmissionLanguageCode|SourceSystemId|
SourceSystemOwner|Source|SourceMedium
METADATA|QuestionnaireParticipant|RequisitionNumber|CandidateNumber|QuestionnaireCode|QstnrVersionNum|
FeedbackResponseFlag|RespondentPersonNumber|Rating
METADATA|QuestionnaireResponse|AttemptNum|QstnrVersionNum|RequisitionNumber|CandidateNumber|
QuestionnaireCode|FeedbackResponseFlag|RespondentPersonNumber|FeedbackCompletionDate
METADATA|QuestionResponse|QuestionId|QuestionCode|QstnVersionNum|AttemptNum|QstnrVersionNum|
QuestionnaireCode|RequisitionNumber|CandidateNumber|SectionSeqNum|AnswerCode
|FeedbackResponseFlag|RespondentPersonNumber

MERGE|CandidateJobApplication|HDLJobRequisition|CAN001|KO|SUB-ZFRCE-HDLJOBREQ|HRC_SQLLOADER|email|campaign
MERGE|QuestionnaireParticipant|44|47001|300000052162209|1|Y|212|5
MERGE|QuestionnaireResponse|1|1|44|47001|300000052162209|Y|212|2023/08/22 21:42:20
MERGE|QuestionResponse|300000049666199|WK_UI_AQ|1|1|1|300000052162209|44|47001|10192|300000049666212|Y|212
```

Deleting Candidate Job Applications

This sample CandidateJobApplication.dat file shows how to delete a candidate job application using the `InactivateFlag` attribute.

```
METADATA|CandidateJobApplication|RequisitionNumber|CandidateNumber|InactivateFlag

MERGE|CandidateJobApplication|HDLJobRequisition|CAN001|Y
```

Deleting Attachments

This sample CandidateJobApplication.dat file shows how to delete an attachment.

```
METADATA|Attachment|AttachedDocumentId|PersonId|SubmissionId
DELETE|Attachment|30000023217|4560700|8945600000

METADATA|Attachment|Title|File|DataTypeCode|URLorFileName|RequisitionNumber|CandidateNumber|Category
DELETE|Attachment|CAN001-Resume-File|Resume.rtf|FILE|Resume.rtf|HDLJobRequisition|CAN001|
IRC_CANDIDATE_RESUME
```

Movement of Job Applications Through Phases and States

Movement Before the Offer Phase

This sample CandidateJobApplication.dat file shows how to move a candidate job application through the Candidate Selection Process prior to the Offer phase.

```
METADATA|CandidateJobApplication|RequisitionNumber|CandidateNumber|CurrentPhaseCode|CurrentStateCode

MERGE|CandidateJobApplication|HDLJobRequisition|CAN001|NEW|UNDER_CONSIDERATION
MERGE|CandidateJobApplication|HDLJobRequisition|CAN002|INTERVIEW|IN_PROGRESS
```

Movement in the Offer Phase

This sample CandidateJobApplication.dat file shows how to move a candidate job application between two different states in the Offer phase.

```
METADATA|CandidateJobApplication|RequisitionNumber|CandidateNumber|CurrentPhaseCode|CurrentStateCode

MERGE|CandidateJobApplication|HDLJobRequisition|CAN001|OFFER|OFFER_EXTENDED
MERGE|CandidateJobApplication|HDLJobRequisition|CAN002|OFFER|OFFER_ACCEPTED
```

Movement in the HR Phase

This sample CandidateJobApplication.dat file shows how to move a candidate job application to the Pending Automated Processing and Rejected by Employer states in the HR phase.

```
METADATA|CandidateJobApplication|RequisitionNumber|CandidateNumber|CurrentPhaseCode|CurrentStateCode

MERGE|CandidateJobApplication|HDLJobRequisition|CAN001|HR|AUTOPROCESSING_PENDING
MERGE|CandidateJobApplication|HDLJobRequisition|CAN002|HR|REJECTED_EMPLOYER
```

Candidate Pool

Overview of Loading Candidate Pool

You can load the candidate pool details comprising the candidate pool member and its pool interaction, the candidate pool owner, and talent community details using the HCM Data Loader (HDL).

Guidelines for Loading Candidate Pool

This topic describes the considerations for loading candidate pool using HCM Data Loader (HDL).

- You should assign a minimum of two owners for a candidate pool having **OwnershipType** as **Shared**.
- You should assign only one owner for a private pool.

Examples of Loading Candidate Pool

This topic provides examples that show how to load candidate pools.

Creating Candidate Pools

This example CandidatePool.dat file creates private, shared, and global candidate pools and adds candidates to it.

```
METADATA|CandidatePool|PoolName|PoolTypeCode|Description|Status|SourceSystemOwner|SourceSystemId|
OwnerPersonNumber|JobSetCode|JobCode|DepartmentName|OwnershipType|SourceSystemId|SourceSystemOwner
MERGE|CandidatePool|POOL-1|CANDIDATE|POOL_Private|A|HRC_SQLLOADER|POOL-ID1|POOLOWNER01|COMMON|ZFRCE ZB
 Analyst|ZFRCE Executive Office|ORA_PRIVATE|POOL-1|HRC_SQLLOADER
MERGE|CandidatePool|POOL-2|CANDIDATE|POOL_Shared|A|HRC_SQLLOADER|POOL-ID2|POOLOWNER01|COMMON|ZFRCE ZB
 Analyst|ZFRCE Executive Office|ORA_SHARED|POOL-2|HRC_SQLLOADER
MERGE|CandidatePool|POOL-3|CANDIDATE|POOL_Global|A|HRC_SQLLOADER|POOL-ID3|POOLOWNER01|COMMON|ZFRCE ZB
 Analyst|ZFRCE Executive Office|ORA_PUBLIC|POOL-3|HRC_SQLLOADER
```

```
METADATA|CandidatePoolOwner|FavoriteFlag|OwnerPersonNumber|PoolName|PoolStatus|SourceSystemId|
SourceSystemOwner
MERGE|CandidatePoolOwner|Y|ZFRCE071_ZBEN|POOL-2|A|POOLOWNER01|HRC_SQLLOADER
METADATA|CandidatePoolMember|CurrentPhaseId|CurrentStateId|CandidateNumber|PoolName|Status|PoolMemberStatus|
SourceSystemId|SourceSystemOwner
MERGE|CandidatePoolMember|55|551|CAN001|POOL-1|A|ACTIVE|POOL-MEMID-1|HRC_SQLLOADER
MERGE|CandidatePoolMember|66|662|CAN002|POOL-1|A|ACTIVE|POOL-MEMID-2|HRC_SQLLOADER
MERGE|CandidatePoolMember|55|551|CAN002|POOL-2|A|ACTIVE|POOL-MEMID-3|HRC_SQLLOADER
MERGE|CandidatePoolMember|66|662|CAN003|POOL-2|A|ACTIVE|POOL-MEMID-4|HRC_SQLLOADER
MERGE|CandidatePoolMember|55|551|CAN003|POOL-3|A|ACTIVE|POOL-MEMID-5|HRC_SQLLOADER
```

Adding Candidates to an existing Candidate Pool

This example CandidatePool.dat file adds a candidate to an existing candidate pool.

```
METADATA|CandidatePoolMember|CurrentPhaseId|CurrentStateId|CandidateNumber|PoolName|Status|PoolMemberStatus|
SourceSystemId|SourceSystemOwner
MERGE|CandidatePoolMember|55|551|CAN001|POOL-3|A|ACTIVE|POOL-MEMID-6|HRC_SQLLOADER
```

Deleting Candidates from Candidate Pool

This example CandidatePool.dat file deletes a candidate from a candidate pool.

```
METADATA|CandidatePoolMember|CurrentPhaseId|CurrentStateId|CandidateNumber|PoolName|Status|PoolMemberStatus|
SourceSystemId|SourceSystemOwner
DELETE|CandidatePoolMember|55|551|CAN002|POOL-1||ACTIVE|POOL-MEMID-2|HRC_SQLLOADER
```

Geography Hierarchy

Overview of Loading Geography Hierarchy

A geography hierarchy object comprises of geography hierarchy and geography hierarchy node in a parent-child relationship. The geography hierarchy node represents a location added to the geography hierarchy. They must be loaded together.

Guidelines for Loading Geography Hierarchy

This topic describes the considerations for loading geography hierarchy using HCM Data Loader (HDL).

Geography Hierarchy Data

User Key

The user key for geography hierarchy is Name.

Name

You can activate a geography hierarchy by providing a valid geography hierarchy node as per its current structure.

Start Date and StartOnActivationFlag

You must provide a unique StartDate. If the StartOnActivationFlag is Y, then don't provide a StartDate. If it's N, then you can set a future StartDate.

StatusCode

You can set only one geography hierarchy as current by setting the StatusCode as `ORA_ACTIVE` and the StartDate to the most recent past date.

> **Note:** You can't delete a current geography hierarchy. Node names are validated against the TCA geography reference data.

Geography Hierarchy Node Data

You can define the geography hierarchy structure for various countries using the geography hierarchy structure FSM. You can select a maximum of two sub-levels per country.

For example, you may configure the two sub-levels for the United States in the following ways:

- Level1 - State, Level2 - City
- Level1 - County, Level2 - City

Given below is the geography structure for various countries used in GeographyHierarchy.dat.

Country Name	Country Code	Level 1	Level 2
United States	US	State	City
Italy	IT	Regione	Provincia
Canada	CA	Province	City
Germany	DE	-	-
France	FR	Region	Commune

User Key

User keys for geography hierarchy node are listed below.

- Name, CountryCode
- Name, Level1, CountryCode
- Name, Level2, Level1, CountryCode
- Name, Level1, CountryCode, GeographyElement1, GeographyElement2
- Name, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3
- Name, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4
- Name, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5
- Name, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5, GeographyElement6
- Name, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5, GeographyElement6, GeographyElement7

- Name, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5, GeographyElement6, GeographyElement7, GeographyElement8
- Name, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5, GeographyElement6, GeographyElement7, GeographyElement8, GeographyElement9
- Name, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5, GeographyElement6, GeographyElement7, GeographyElement8, GeographyElement9, GeographyElement10
- Name, Level2, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3
- Name, Level2, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4
- Name, Level2, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5
- Name, Level2, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5, GeographyElement6
- Name, Level2, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5, GeographyElement6, GeographyElement7
- Name, Level2, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5, GeographyElement6, GeographyElement7, GeographyElement8
- Name, Level2, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5, GeographyElement6, GeographyElement7, GeographyElement8, GeographyElement9
- Name, Level2, Level1, CountryCode, GeographyElement1, GeographyElement2, GeographyElement3, GeographyElement4, GeographyElement5, GeographyElement6, GeographyElement7, GeographyElement8, GeographyElement9, GeographyElement10

You can primarily use the first three of the user keys viz. Name, CountryCode, or Name, Level1, CountryCode, or Name, Level2, Level1, CountryCode.

You can use the remaining seventeen if there are two locations with the same user key combinations as in Level1, CountryCode, or Level2, Level1, CountryCode. This is because a location in TCA geographies can have up to ten levels. So, you must provide all the geography elements to identify the location uniquely.

Examples of Loading Geography Hierarchy

This topic provides examples that show how to load geography hierarchies.

Creating Geography Hierarchy

This example GeographyHierarchy.dat file creates a basic geography hierarchy record.

```
METADATA|GeographyHierarchy|Name|StartDate|StartOnActivationFlag|StatusCode
MERGE|GeographyHierarchy|HDLGH-Jun25-0544||Y|ORA_ACTIVE
METADATA|GeographyHierarchyNode|Name|HierarchyId|CountryCode|Level1|Level2|GeographyElement1|
GeographyElement2|GeographyElement3|GeographyElement4|GeographyElement5|GeographyElement6|GeographyElement7|
GeographyElement8|GeographyElement9|GeographyElement10|DeleteFlag
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0544||DE|||||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0544||IT|TOSCANA|FIRENZE||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0544||CA|NL||||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0544||CA|ON|Toronto||||||||||
```

```
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0544||CA|ON|Mississauga||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0544||US|WA|SEATTLE||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0544||US|SC|||||||||||
```

In this GeographyHierarchy.dat, for CountryCode 'US,' there are two records with the values WI, ABBOTSFORD, United States, WI at level 1, level 2, geography element 1, geography element 2, respectively. Hence, by additionally configuring the values CLARK, ABBOTSFORD, and MARATHON, ABBOTSFORD at geography element 3, and geography element 4 for the records, you can uniquely define two geography nodes.

```
METADATA|GeographyHierarchy|Name|StartDate|StartOnActivationFlag|StatusCode
MERGE|GeographyHierarchy|HDLGH-Jun25-0547|2021/06/29 00:00:00|N|ORA_DRAFT
METADATA|GeographyHierarchyNode|Name|HierarchyId|CountryCode|Level1|Level2|GeographyElement1|
GeographyElement2|GeographyElement3|GeographyElement4|GeographyElement5|GeographyElement6|GeographyElement7|
GeographyElement8|GeographyElement9|GeographyElement10|DeleteFlag
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||DE|||||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||IT|TOSCANA|FIRENZE||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||CA|NL|||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||CA|ON|Toronto||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||CA|ON|Mississauga||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|WA|SEATTLE||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|SC|||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|WI|ABBOTSFORD|United States|WI|CLARK|ABBOTSFORD|||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|WI|ABBOTSFORD|United States|WI|MARATHON|ABBOTSFORD|||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|MD|Parkville|United States|MD|Baltimore (Ind City)|
Parkville|||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|MD|Parkville|United States|MD|Baltimore|Parkville|||||||
```

Updating Geography Hierarchy

This example GeographyHierarchy.dat file updates the StartDate and StatusCode of the geography hierarchy, adds a new geography hierarchy node for France, and deletes the geography hierarchy node for Germany.

```
METADATA|GeographyHierarchy|Name|StartDate|StartOnActivationFlag|StatusCode
MERGE|GeographyHierarchy|HDLGH-Jun25-0547|2021/06/30 00:00:00|N|ORA_ACTIVE
METADATA|GeographyHierarchyNode|Name|HierarchyId|CountryCode|Level1|Level2|GeographyElement1|
GeographyElement2|GeographyElement3|GeographyElement4|GeographyElement5|GeographyElement6|GeographyElement7|
GeographyElement8|GeographyElement9|GeographyElement10|DeleteFlag
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||DE|||||||||||||Y
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||IT|TOSCANA|FIRENZE||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||CA|NL|||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||CA|ON|Toronto||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||CA|ON|Mississauga||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|WA|SEATTLE||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|SC|||||||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|WI|ABBOTSFORD|United States|WI|CLARK|ABBOTSFORD|||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|WI|ABBOTSFORD|United States|WI|MARATHON|ABBOTSFORD|||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|MD|Parkville|United States|MD|Baltimore (Ind City)|
Parkville|||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||US|MD|Parkville|United States|MD|Baltimore|Parkville|||||||
MERGE|GeographyHierarchyNode|HDLGH-Jun25-0547||FR|||||||||||||
```

Deleting Geography Hierarchy

This example GeographyHierarchy.dat file deletes the geography hierarchy record using the Name attribute.

```
METADATA|GeographyHierarchy|Name|StartDate|StartOnActivationFlag|StatusCode
DELETE|GeographyHierarchy|HDLGH-Jun25-0547|||
```

Job Referrals

Overview of Loading Job Referrals

Using HCM Data Loader (HDL), you can load job referrals for external and internal candidates by employees, and job referrals for external candidates by agents. You can also load attachments for job referrals for external candidates by employees.

Guidelines for Loading Job Referrals

This topic describes the considerations for loading job referrals.

General Considerations

These are the general considerations for loading job referrals:

- When loading a referral using HCM Data Loader, a record is created internally for a prospect of type, **Referred**.
- Source tracking information is also created internally.
- The user can't refer a candidate to a job requisition that isn't in the Open phase.
- External or internal candidates can be referred depending on where the requisition is posted. For example, if a requisition is posted internally, then only internal candidates can be referred. If a requisition is posted both internally and externally, then both internal and external candidates can be referred.

Supported Actions

Using HDL, you can create and delete job referrals. You can create attachments only for external candidates referred by employees. You can't delete the attachments using HDL.

Referrals by Employees for External Candidates

User Key

You must provide both these user keys while loading an external candidate referral:

- RequisitionNumber
- CandidateNumber

Referrals by Employees for Internal Candidates

User Key

You must provide both these user keys while loading an internal candidate referral:

- RequisitionNumber
- PersonNumber

Referrals by Agencies for External Candidates

General Considerations

- The agent must be invited to submit candidates to the job requisition.
- The candidate referred by the agent should be a part of the talent pool of the agency.

User Key

You must provide both these user keys while loading an external candidate referral:

- RequisitionNumber
- CandidateNumber

Agent Name

This must be the first name and last name of the agent.

Attachments

User Key

You must provide all these user keys when loading attachments for referrals:

- DataTypeCode, Title, RequisitionNumber, PersonNumber
- DataTypeCode, Title, RequisitionNumber, CandidateNumber

Attachment Category

Specify this as `IRC_CANDIDATE_RESUME` in the .dat file.

Examples for Loading Job Referrals

This topic provides sample files that show how to load referrals and attachments.

Creating or Updating Referrals

Referrals by Employees for External Candidates

This example Referral.dat file creates a basic referral record with the attributes RequisitionNumber, CandidateNumber, NotesCandidate, and ReferrerPersonNumber.

```
METADATA|Referral|NotesCandidate|RequisitionNumber|CandidateNumber|ReferrerPersonNumber

MERGE|Referral|Test Referral|ZBEN_Test BGC_PUNCHOUT|264884|ZFRCE001_ZBEN
```

Referrals by Employees for Internal Candidates

This example Referral.dat file creates a basic referral record with the attributes RequisitionNumber, PersonNumber, NotesCandidate, and ReferrerPersonNumber.

```
METADATA|Referral|NotesCandidate|RequisitionNumber|PersonNumber|ReferrerPersonNumber

MERGE|Referral|Test Referral|ZBEN_Test BGC_PUNCHOUT|300100010473106|ZFRCE001_ZBEN
```

Referrals by Agencies for External Candidates

This example Referral.dat file creates a basic referral record with the attributes RequisitionNumber, CandidateNumber, AgencyName, AgentName, and NotesCandidate.

```
METADATA|Referral|NotesCandidate|RequisitionNumber|CandidateNumber|AgencyName|AgentName

MERGE|Referral|Test Referral|ZBEN_Test BGC_PUNCHOUT|235553|ABC IT Recruiting Agency|Das Das
```

Deleting Referrals

The same user key combinations that are used for creating referrals must be used for deleting referrals, based on internal or external candidates.

This sample file deletes a referral for an external candidate by an employee.

```
METADATA|Referral|NotesCandidate|RequisitionNumber|CandidateNumber|ReferrerPersonNumber

DELETE|Referral|Test Referral|ZBEN_Test BGC_PUNCHOUT|264884|ZFRCE001_ZBEN
```

Creating or Updating Attachments

This example Referral.dat file uploads an attachment to a referral with attributes such as Title, RequisitionNumber, CandidateNumber, Category, and URLorFileName.

```
METADATA|Attachment|Title|File|DataTypeCode|URLorFileName|Category|RequisitionNumber|CandidateNumber

MERGE|Attachment|HDLAttachment1|HDLAttachment1.txt|FILE|HDLAttachment1.txt|IRC_CANDIDATE_RESUME|ZBEN_Test
 BGC_PUNCHOUT|264884
```

Job Requisitions

Overview of Loading Job Requisitions

Using the HCM Data Loader (HDL), you can create new job requisitions in various states and phases, update job requisitions, and set job requisitions to the Deleted status.

When updating job requisitions, you can change their state and phase, and update certain details depending on their current state or phase.

Guidelines for Loading Job Requisitions

This topic describes the general considerations for loading job requisitions using HDL. It also lists the states and phases in which you can create and update job requisitions using HDL.

General Considerations

- If the assignment of the recruiter and/or hiring manager is not available in the job load file, the primary assignment is automatically used during the load.

- The content for all descriptions, for example, `InternalDescription`, `InternalShortDescription`, etc., must be provided in a .txt file within the ClobFiles folder. This text file can contain HTML content for rich formatting.
- The `BaseLanguageCode` in the JobRequisition line should be one of the available active languages in the application. This is a mandatory field. It determines the base language of the job requisition and is used for translations.
- `MediaLinkLanguage` has to be loaded along with the MediaLink line and cannot be loaded without the parent MediaLink line.
- When you create or update a job requisition in the Open phase and Posted state, you must provide at least one PostingDetails line with these considerations:
  - The posting start date must be after the job requisition creation date.
  - `PostingStatus` is auto-populated based on the posting start and end dates.

Considerations for Creating Job Requisitions

Using HDL, you can create a job requisition directly in a specific phase and state. However, you must provide all the mandatory information required for that phase or state.

You can create job requisitions directly in these phase and state combinations:

PHASE CODE	STATE CODE
REQUISITION_DRAFT	• REQUISITION_IN_PROGRESS • REQUISITION_DELETED
REQUISITION_APPROVAL	• REQUISITION_REJECTED • REQUISITION_CANCELLED
REQUISITION_JOB_FORMATTING	• REQUISITION_IN_PROGRESS • REQUISITION_CANCELLED
REQUISITION_POSTING	• REQUISITION_IN_PROGRESS • REQUISITION_CANCELLED
REQUISITION_OPEN	• REQUISITION_NOT_POSTED • REQUISITION_NOT_POSTED • REQUISITION_POSTED • REQUISITION_SCHEDULED • REQUISITION_UNPOSTED • REQUISITION_EXPIRED • REQUISITION_CANCELLED • REQUISITION_FILLED

Considerations for Updating Job Requisitions

Using HDL, you can move job requisitions from one phase and state to another phase and state, subject to certain restrictions.

Note: If a job requisition is in the Canceled or Deleted state, you can't update it.

This table shows the various phases and states to which you can move a job requisition:

CURRENT PHASE - STATE CODE	DESTINATION PHASE - STATE CODE
REQUISITION_DRAFT - REQUISITION_IN_PROGRESS	• REQUISITION_DRAFT - REQUISITION_DELETED • REQUISITION_JOB_FORMATTING - REQUISITION_IN_PROGRESS • REQUISITION_POSTING - REQUISITION_IN_PROGRESS • REQUISITION_OPEN - REQUISITION_NOT_POSTED • REQUISITION_OPEN - REQUISITION_POSTED • REQUISITION_OPEN - REQUISITION_SCHEDULED • REQUISITION_OPEN - REQUISITION_UNPOSTED • REQUISITION_OPEN - REQUISITION_EXPIRED
REQUISITION_DRAFT - REQUISITION_DELETED	• None
REQUISITION_APPROVAL - REQUISITION_PENDING	• None
REQUISITION_APPROVAL - REQUISITION_REJECTED	• REQUISITION_DRAFT - REQUISITION_IN_PROGRESS • REQUISITION_APPROVAL - REQUISITION_CANCELLED
REQUISITION_APPROVAL - REQUISITION_CANCELLED	• REQUISITION_APPROVAL - REQUISITION_REJECTED
REQUISITION_JOB_FORMATTING - REQUISITION_CANCELLED	• REQUISITION_JOB_FORMATTING - REQUISITION_IN_PROGRESS
REQUISITION_JOB_FORMATTING - REQUISITION_IN_PROGRESS	• REQUISITION_DRAFT - REQUISITION_IN_PROGRESS • REQUISITION_JOB_FORMATTING - REQUISITION_CANCELLED • REQUISITION_POSTING - REQUISITION_IN_PROGRESS • REQUISITION_OPEN - REQUISITION_NOT_POSTED • REQUISITION_OPEN - REQUISITION_POSTED • REQUISITION_OPEN - REQUISITION_SCHEDULED • REQUISITION_OPEN - REQUISITION_UNPOSTED • REQUISITION_OPEN - REQUISITION_EXPIRED
REQUISITION_POSTING - REQUISITION_CANCELLED	• REQUISITION_POSTING - REQUISITION_IN_PROGRESS
REQUISITION_POSTING - REQUISITION_IN_PROGRESS	• REQUISITION_DRAFT - REQUISITION_IN_PROGRESS • REQUISITION_POSTING - REQUISITION_CANCELLED • REQUISITION_OPEN - REQUISITION_NOT_POSTED • REQUISITION_OPEN - REQUISITION_POSTED • REQUISITION_OPEN - REQUISITION_SCHEDULED • REQUISITION_OPEN - REQUISITION_UNPOSTED

CURRENT PHASE - STATE CODE	DESTINATION PHASE - STATE CODE
	• REQUISITION_OPEN - REQUISITION_EXPIRED
REQUISITION_OPEN - REQUISITION_CANCELLED	• REQUISITION_OPEN - REQUISITION_NOT_POSTED • REQUISITION_OPEN - REQUISITION_UNPOSTED
REQUISITION_OPEN - REQUISITION_EXPIRED	• REQUISITION_DRAFT - REQUISITION_IN_PROGRESS • REQUISITION_OPEN - REQUISITION_POSTED • REQUISITION_OPEN - REQUISITION_SCHEDULED • REQUISITION_OPEN - REQUISITION_CANCELLED • REQUISITION_OPEN - REQUISITION_FILLED
REQUISITION_OPEN - REQUISITION_FILLED	• REQUISITION_OPEN - REQUISITION_NOT_POSTED • REQUISITION_OPEN - REQUISITION_UNPOSTED
REQUISITION_OPEN - REQUISITION_NOT_POSTED	• REQUISITION_DRAFT - REQUISITION_IN_PROGRESS • REQUISITION_OPEN - REQUISITION_POSTED • REQUISITION_OPEN - REQUISITION_SCHEDULED • REQUISITION_OPEN - REQUISITION_CANCELLED • REQUISITION_OPEN - REQUISITION_FILLED • REQUISITION_OPEN - REQUISITION_SUSPENDED
REQUISITION_OPEN - REQUISITION_POSTED	• REQUISITION_DRAFT - REQUISITION_IN_PROGRESS • REQUISITION_OPEN - REQUISITION_SCHEDULED • REQUISITION_OPEN - REQUISITION_CANCELLED • REQUISITION_OPEN - REQUISITION_FILLED • REQUISITION_OPEN - REQUISITION_SUSPENDED • REQUISITION_OPEN - REQUISITION_UNPOSTED
REQUISITION_OPEN - REQUISITION_SCHEDULED	• REQUISITION_DRAFT - REQUISITION_IN_PROGRESS • REQUISITION_OPEN - REQUISITION_POSTED • REQUISITION_OPEN - REQUISITION_CANCELLED • REQUISITION_OPEN - REQUISITION_FILLED • REQUISITION_OPEN - REQUISITION_SUSPENDED • REQUISITION_OPEN - REQUISITION_UNPOSTED
REQUISITION_OPEN - REQUISITION_SUSPENDED	• REQUISITION_DRAFT - REQUISITION_IN_PROGRESS • REQUISITION_OPEN - REQUISITION_UNPOSTED • REQUISITION_OPEN - REQUISITION_NOT_POSTED • REQUISITION_OPEN - REQUISITION_CANCELLED • REQUISITION_OPEN - REQUISITION_FILLED
REQUISITION_OPEN - REQUISITION_UNPOSTED	• REQUISITION_DRAFT - REQUISITION_IN_PROGRESS • REQUISITION_OPEN - REQUISITION_POSTED

CURRENT PHASE - STATE CODE	DESTINATION PHASE - STATE CODE
	• REQUISITION_OPEN - REQUISITION_SCHEDULED • REQUISITION_OPEN - REQUISITION_CANCELLED • REQUISITION_OPEN - REQUISITION_FILLED • REQUISITION_OPEN - REQUISITION_SUSPENDED

User Key

You must provide this user key while loading a job requisition:

- RequisitionNumber

Examples of Loading Job Requisitions

This topic provides sample files that show how to create and update job requisitions. It also shows how to set a job requisition to the Deleted status.

Creating Job Requisitions

Job Requisition in Draft – In Progress

This sample JobRequisition.dat file creates a basic job requisition in the Draft – In Progress status using the RequisitionNumber, CurrentPhaseCode, and CurrentStateCode, PrimaryLocationName, and OrganizationCode attributes.

```
METADATA|JobRequisition|RequisitionNumber|RecruitingType|UnlimitedOpeningsFlag|RequisitionTitle|
HiringManagerPersonNumber|RecruiterPersonNumber|PrimaryLocationName|OrganizationCode|CurrentPhaseCode|
CurrentStateCode|PrimaryWorkLocationCode|BaseLanguageCode|CandidateSelectionProcessCode|
ExternalApplyFlowCode|SourceSystemId|SourceSystemOwner

MERGE|JobRequisition|HDLJobRequisition|ORA_CAMPUS|Y|HDL-JobRequisition-Title|ZFRCE0571_ZBEN|ZFRCE001_ZBEN|
Brazil|ZBEN United Benefits Corporation|REQUISITION_DRAFT|REQUISITION_IN_PROGRESS|ZFRCELOC1|US|CSP-DEFAULT|
DEFAULT_AF|HDL-JobRequisition-SysID-1|HRC_SQLLOADER
```

Job Requisition with Child Objects

This sample JobRequisition.dat file creates a job requisition in the Open – Posted status with these child objects: PostingDetails, Language, Hiring Team, Other Location, Other Work Location, and Job Profile. The sample file below contains all the mandatory attributes required for the job load.

```
METADATA|JobRequisition|RequisitionNumber|RecruitingType|NumberOfOpenings|UnlimitedOpeningsFlag|
RequisitionTitle|InternalRequisitionTitle|BusinessJustification|JobFunction|SourcingBudget|
TravelBudget|RelocationBudget|EmployeeReferralBonus|MaximumSalary|MinimumSalary|PayFrequency|WorkerType|
RegularOrTemporary|ManagementLevel|FullTimeOrPartTime|JobShift|JobType|EducationLevel|ContactNameExternal|
ContactEmailExternal|ExternalShortDescription|ExternalDescription|ExternalEmployerDescriptionId|
ExternalOrganizationDescriptionId|ContactNameInternal|ContactEmailInternal|InternalShortDescription|
InternalDescription|InternalEmployerDescriptionId|InternalOrganizationDescriptionId|DisplayInOrgChartFlag|
Comments|SourceSystemOwner|SourceSystemId|JobCode|HiringManagerPersonNumber|RecruiterPersonNumber|
PrimaryLocationName|GradeCode|OrganizationName|ClassificationCode|JobFamilyName|LegalEmployerName|
BusinessUnitShortCode|DepartmentName|CurrentPhaseCode|CurrentStateCode|PrimaryWorkLocationCode|
BudgetCurrencyName|SalaryCurrencyName|BaseLanguageCode|CandidateSelectionProcessCode|ExternalApplyFlowCode

MERGE|JobRequisition|HDL-JobRequisitionOpenPosted|ORA_EXECUTIVES|6|No|HDL-JobRequisitionOpenPosted-Title|
HDL-JobRequisitionOpenPosted-InternalTitle|New Position|Managerial|11000|12000|13000|14000|15000000|
13000000|Annually|Employee|Regular|Supervisor|Full Time|Night|Standard|First Professional|External Name|
```

```
external@gmail.com|ExtShortDesc-EN.txt|ExtLongDesc-EN.txt|300100126268471|300100171064623|Internal name|
internal@gmail.com|IntShortDesc-EN.txt|IntLongDesc-EN.txt|300100171064619|300100171064623|No|Comments2.txt|
HRC_SQLLOADER|HDL-JobRequisitionOpenPosted-SysID|ZFRCE CHRO|ZFRCE0525_ZBEN|ZFRCE001_ZBEN|United States|
ZFRCE MGMT3|Vision Corporation Enterprise, ZBEN United Benefits Corporation|DEPARTMENT|ZFRCE_Management|
ZFRCE_US_LE1_ZBEN|ZBEN_Common_Business Unit|ZFRCE US Payables|REQUISITION_OPEN|REQUISITION_POSTED|ZFRCELOC1|
US Dollar|US Dollar|US|HDLTestCSP-001|ORA_GLOBAL_EXTERNAL

METADATA|PostingDetails|RequisitionNumber|ExternalOrInternal|StartDate|EndDate

MERGE|PostingDetails|HDL-JobRequisitionOpenPosted|Internal|2018/02/03 00:00:00|2030/12/11 00:00:00

MERGE|PostingDetails|HDL-JobRequisitionOpenPosted|External|2018/02/03 00:00:00|2030/12/11 00:00:00

METADATA|Language|LanguageCode|RequisitionNumber|SourceSystemId|SourceSystemOwner

MERGE|Language|KO|HDL-JobRequisitionOpenPosted|HDL-JobRequisitionOpenPosted-LANG-1|HRC_SQLLOADER

METADATA|HiringTeam|RequisitionNumber|PersonNumber|MemberType

MERGE|HiringTeam|HDL-JobRequisitionOpenPosted|ZFRCE071_ZBEN|Collaborator

METADATA|OtherLocation|RequisitionNumber|OtherLocationName|SourceSystemId|SourceSystemOwner

MERGE|OtherLocation|HDL-JobRequisitionOpenPosted|Germany|HDL-JobRequisitionOpenPosted-OLOC-1|HRC_SQLLOADER

METADATA|OtherWorkLocation|RequisitionNumber|OtherWorkLocationCode

MERGE|OtherWorkLocation|HDL-JobRequisitionOpenPosted|ZBEN_Greece

METADATA|JobProfile|RequisitionNumber|SourceSystemOwner|SourceSystemId|ContentType|DateFrom|ItemText301|
ItemText303|ItemText305|ItemText3014|ItemText3015|ItemNumber1|ItemNumber2

MERGE|JobProfile|HDL-JobRequisitionOpenPosted|HRC_SQLLOADER|HDL-JobRequisitionOpenPosted-JP-1|
WORK_REQUIREMENTS|2016/09/27|Y|N|N|MTS|WH1|7|5
```

Adding a Questionnaire and Questions to a Job Requisition

This sample JobRequisition.dat file adds an additional questionnaire and question to a job requisition. Prescreening questionnaires, if defined, are defaulted based on the context. The sample file below contains all the mandatory attributes required for the load.

```
METADATA|InterviewQuestionnaire|RequisitionNumber|QuestionnaireCode|TemplateFlag|QstnrVersionNum

MERGE|InterviewQuestionnaire|HDL-JobRequisition|300100166479930|N|1

METADATA|ApplicationSpecificQuestion|RequisitionNumber|QstnrTypeCode|QuestionCode|QstnVersionNum|AdhocFlag

MERGE|ApplicationSpecificQuestion|HDL-JobRequisition|ORA_INTERNAL_APPLYFLOW_QSTNR|500110-0278|1|N
```

Adding a Partner Configuration to the Job Requisition

This sample JobRequisition.dat file adds the background check, tax credit, and assessment child objects to a job requisition. The sample file below contains all the mandatory attributes required for the load.

Background Check

```
METADATA|BackgroundCheckConfiguration|RequisitionNumber|PartnerName|AccountUserName|MultipleRequestFlag|
GUID|AccountId|ReqAccountId|AllowUpdateOnJaFlag|ProvisioningId|RequisitionId|SourceSystemId|
SourceSystemOwner

MERGE|BackgroundCheckConfiguration|HDL-JobRequisition|HDLPartnerB221|BGCBrnzU|Y||||||CFGSSID1|HRC_SQLLOADER

METADATA|BackgroundCheckScreeningPackageV2|RequisitionNumber|PartnerName|AccountUserName|ScrPkgCode|
ScrPkgName|AddedBy|EnteringPhaseFlag|LeavingPhaseFlag|PhaseCode|StateCode|SourceSystemId|SourceSystemOwner
```

```
MERGE|BackgroundCheckScreeningPackageV2|HDL-JobRequisition|HDLPartnerB221|BGCBrnzU|HPKG1|HDLPkg1|
ZFRCE001_ZBEN|Y|N|NEW||PKGSSID1|HRC_SQLLOADER

MERGE|BackgroundCheckScreeningPackageV2|HDL-JobRequisition|HDLPartnerB221|BGCBrnzU|HPKG2|HDLPkg2|
ZFRCE001_ZBEN|||NEW|TO_BE_REVIEWED|PKGSSID2|HRC_SQLLOADER

MERGE|BackgroundCheckScreeningPackageV2|HDL-JobRequisition|HDLPartnerB221|BGCBrnzU|HPKG3|HDLPkg3|
ZFRCE001_ZBEN|Y||300100539005663||PKGSSID2|HRC_SQLLOADER

MERGE|BackgroundCheckScreeningPackageV2|HDL-JobRequisition|HDLPartnerB221|BGCBrnzU|HPKG4|HDLPkg4|
ZFRCE001_ZBEN|||300100539005663|300100185138381|PKGSSID3|HRC_SQLLOADER

MERGE|BackgroundCheckScreeningPackageV2|HDL-JobRequisition|HDLPartnerB221|BGCBrnzU|HPKG5|HDLPkg5|
ZFRCE001_ZBEN|||SCREENING|PHONE_SCREEN_COMPLETED|PKGSSID4|HRC_SQLLOADER
```

Tax Credit

```
METADATA|TaxCreditConfiguration|AccountUserName|PartnerName|RequisitionNumber|SourceSystemId|
SourceSystemOwner

MERGE|TaxCreditConfiguration|User-PTR2|Tax Credit-PTR|HDL-JobRequisition1|CFGSSIDTC1|HRC_SQLLOADER

MERGE|TaxCreditConfiguration|User-PTR2|Tax Credit-PTR|HDL-JobRequisition2|CFGSSIDTC2|HRC_SQLLOADER

METADATA|TaxCreditScreeningPackage|TriggerTypeCode|AccountUserName|PartnerName|RequisitionNumber|
PackageCode|PackageName|SourceSystemOwner|SourceSystemId

MERGE|TaxCreditScreeningPackage|ORA_IRC_CSP|User-PTR2|Tax Credit-PTR|HDL-JobRequisition1|TCPKGCODE1|
TCPKGNAME1|TCPKGSSID1|HRC_SQLLOADER

MERGE|TaxCreditScreeningPackage|ORA_IRC_APPLY_EXT|User-PTR2|Tax Credit-PTR|HDL-JobRequisition2|TCPKGCODE1|
TCPKGNAME1|TCPKGSSID2|HRC_SQLLOADER
```

Assessment

```
METADATA|AssessmentConfiguration|AccountUserName|PartnerName|RequisitionNumber|MultiPhaseCspFlag|
SourceSystemId|SourceSystemOwner

MERGE|AssessmentConfiguration|HDLABU221|HDL_BA01|HDLJobRequisition|Y|ASMT-CONF-1|HRC_SQLLOADER

METADATA|AssessmentScreeningPackage|AccountUserName|PartnerName|PackageCode|PackageName|TriggerTypeCode|
PhaseCode|StateCode|RequisitionNumber|PackageSequence|SourceSystemId|SourceSystemOwner

MERGE|AssessmentScreeningPackage|HDLABU221|HDL_BA01|TESTPKG1|TestPkg1|ORA_IRC_CSP|NEW|TO_BE_REVIEWED|
HDLJobRequisition|1|ASMT-PKG-1|HRC_SQLLOADER

MERGE|AssessmentScreeningPackage|HDLABU221|HDL_BA01|TESTPKG2|TestPkg2|ORA_IRC_CSP|SCREENING|
PHONE_SCREEN_COMPLETED|HDLJobRequisition|2|ASMT-PKG-2|HRC_SQLLOADER

MERGE|AssessmentScreeningPackage|HDLABU221|HDL_BA01|TESTPKG3|TestPkg3|ORA_IRC_CSP|OFFER|OFFER_EXTENDED|
HDLJobRequisition|3|ASMT-PKG-3|HRC_SQLLOADER

MERGE|AssessmentScreeningPackage|HDLABU221|HDL_BA01|TESTPKG4|TestPkg4|ORA_IRC_APPLY_INT|||HDLJobRequisition|
4|ASMT-PKG-4|HRC_SQLLOADER

MERGE|AssessmentScreeningPackage|HDLABU221|HDL_BA01|TESTPKG5|TestPkg5|ORA_IRC_APPLY_EXT|||HDLJobRequisition|
5|ASMT-PKG-5|HRC_SQLLOADER
```

Creating a Pipeline or Linked Job Requisition

This sample JobRequisition.dat file creates a pipeline job requisition using attributes such as RequisitionNumber, RequisitionTitle, CurrentPhaseCode, CurrentStateCode, and PipelineRequisitionFlag.

Pipeline Requisition

```
METADATA|JobRequisition|RequisitionNumber|RecruitingType|UnlimitedOpeningsFlag|RequisitionTitle|
HiringManagerPersonNumber|RecruiterPersonNumber|PrimaryLocationName|OrganizationCode|CurrentPhaseCode|
CurrentStateCode|PrimaryWorkLocationCode|BaseLanguageCode|CandidateSelectionProcessCode|
ExternalApplyFlowCode|SourceSystemId|SourceSystemOwner|PipelineRequisitionFlag

MERGE|JobRequisition|HDLJobRequisitionPipeline|ORA_CAMPUS|Y|HDL-JobRequisition-Pipeline|ZFRCE0571_ZBEN|
ZFRCE001_ZBEN|Brazil|ZBEN United Benefits Corporation|REQUISITION_DRAFT|REQUISITION_IN_PROGRESS|ZFRCELOC1|
US|CSP-DEFAULT|DEFAULT_AF|HDL-JobRequisition-SysID-1|HRC_SQLLOADER|Y
```

Linking a Hiring Requisition to a Pipeline Requisition

This sample JobRequisition.dat file links a hiring requisition to a pipeline requisition using attributes such as RequisitionNumber, RequisitionTitle, CurrentPhaseCode, CurrentStateCode, and PipelineRequisitionNumber.

```
METADATA|JobRequisition|RequisitionNumber|RecruitingType|UnlimitedOpeningsFlag|RequisitionTitle|
HiringManagerPersonNumber|RecruiterPersonNumber|PrimaryLocationName|OrganizationCode|CurrentPhaseCode|
CurrentStateCode|PrimaryWorkLocationCode|BaseLanguageCode|CandidateSelectionProcessCode|
ExternalApplyFlowCode|SourceSystemId|SourceSystemOwner|PipelineRequisitionNumber

MERGE|JobRequisition|HDLLinkedHiringRequsition|ORA_CAMPUS|Y|HDL-HDLLinkedHiringRequsition|ZFRCE0571_ZBEN|
ZFRCE001_ZBEN|Brazil|ZBEN United Benefits Corporation|REQUISITION_DRAFT|REQUISITION_IN_PROGRESS|ZFRCELOC1|
US|CSP-DEFAULT|DEFAULT_AF|HDL-JobRequisition-SysID-1|HRC_SQLLOADER|HDLJobRequisitionPipeline
```

Creating a Position-Based Job Requisition

The JobCode attribute can be passed to link a job to a requisition but the required values from the job requisition have to be explicitly provided in the JobRequisition.dat file.

Creating a Job-Based Job Requisition

The PositionCode attribute can be passed to link a job to a requisition but the required values from the job requisition have to be explicitly provided in the JobRequisition.dat file.

Creating a Confidential Job Requisition

This sample JobRequisition.dat file creates a confidential job requisition using attributes such as RequisitionNumber, CurrentPhaseCode, and CurrentStateCode, and ApplyWhenNotPostedFlag.

```
METADATA|JobRequisition|RequisitionNumber|RecruitingType|NumberOfOpenings|ApplyWhenNotPostedFlag|
UnlimitedOpeningsFlag|RequisitionTitle|InternalRequisitionTitle|BusinessJustification|JobFunction|
SourcingBudget|TravelBudget|RelocationBudget|EmployeeReferralBonus|MaximumSalary|MinimumSalary|
PayFrequency|WorkerType|RegularOrTemporary|ManagementLevel|FullTimeOrPartTime|JobShift|
JobType|EducationLevel|ContactNameExternal|ContactEmailExternal|ExternalShortDescription|
ExternalDescription|ExternalEmployerDescriptionId|ExternalOrganizationDescriptionId|ContactNameInternal|
ContactEmailInternal|InternalShortDescription|InternalDescription|InternalEmployerDescriptionId|
InternalOrganizationDescriptionId|DisplayInOrgChartFlag|Comments|SourceSystemOwner|SourceSystemId|
JobCode|HiringManagerPersonNumber|RecruiterPersonNumber|PrimaryLocationName|GradeCode|OrganizationName|
ClassificationCode|JobFamilyName|LegalEmployerName|BusinessUnitShortCode|DepartmentName|CurrentPhaseCode|
CurrentStateCode|PrimaryWorkLocationCode|BudgetCurrencyName|SalaryCurrencyName|BaseLanguageCode|
CandidateSelectionProcessCode|ExternalApplyFlowCode

MERGE|JobRequisition|HDLJobRequisition|EXECUTIVE|7|YES|No|HDLJobRequisition-Title|HDLJobRequisition-
InternalTitle|New Position|Managerial|11000|12000|13000|14000|15000000|13000000|Annually|Employee|Regular|
Supervisor|Full Time|Night|Standard|First Professional|External Name|external@gmail.com|ExtShortDesc-EN.txt|
ExtLongDesc-EN.txt|300100126268471|300100171064623|Internal name|internal@gmail.com|IntShortDesc-EN.txt|
IntLongDesc-EN.txt|300100171064619|300100171064623|No|Comments2.txt|HRC_SQLLOADER|HDLJobRequisition-SysID|
ZFRCE CHRO|ZFRCE0525_ZBEN|ZFRCE001_ZBEN|United States|ZFRCE MGMT3|Vision Corporation Enterprise, ZBEN United
 Benefits Corporation|DEPARTMENT|ZFRCE_Management|ZFRCE_US_LE1_ZBEN|ZBEN_Common_Business Unit|ZFRCE US
 Payables|REQUISITION_OPEN|REQUISITION_NOT_POSTED|ZFRCELOC1|US Dollar|US Dollar|US|HDLTestCSP-001|HDL-
JAFlow01
```

Updating Requisitions

This sample JobRequisition.dat file moves a job requisition to the Job Formatting – In Progress status and updates its primary work location and Candidate Selection Process (CSP) using these attributes: CurrentPhaseCode, CurrentStateCode, PrimaryWorkLocationCode, and CandidateSelectionProcessCode. In addition to these, the RequisitionNumber attribute is mandatory.

```
METADATA|JobRequisition|RequisitionNumber|RecruitingType|UnlimitedOpeningsFlag|RequisitionTitle|
HiringManagerPersonNumber|RecruiterPersonNumber|PrimaryLocationName|OrganizationCode|CurrentPhaseCode|
CurrentStateCode|PrimaryWorkLocationCode|BaseLanguageCode|CandidateSelectionProcessCode|
ExternalApplyFlowCode|SourceSystemId|SourceSystemOwner

MERGE|JobRequisition|HDLJobRequisition|ORA_CAMPUS|Y|HDL-JobRequisition-Title|ZFRCE0571_ZBEN|ZFRCE001_ZBEN|
Brazil|ZBEN United Benefits Corporation|REQUISITION_JOB_FORMATTING|REQUISITION_IN_PROGRESS|ZFRCELOC1UPD|US|
CSP-UPDATED|DEFAULT_AF|HDL-JobRequisition-SysID-1|HRC_SQLLOADER
```

Deleting Requisitions

This sample JobRequisition.dat file moves a job requisition to the Draft – Deleted status by updating the CurrentPhaseCode and CurrentStateCode attributes. In addition to these, the RequisitionNumber attribute is mandatory.

```
METADATA|JobRequisition|RequisitionNumber|RecruitingType|UnlimitedOpeningsFlag|RequisitionTitle|
HiringManagerPersonNumber|RecruiterPersonNumber|PrimaryLocationName|OrganizationCode|CurrentPhaseCode|
CurrentStateCode|PrimaryWorkLocationCode|BaseLanguageCode|CandidateSelectionProcessCode|
ExternalApplyFlowCode|SourceSystemId|SourceSystemOwner

MERGE|JobRequisition|HDLJobRequisition|ORA_CAMPUS|Y|HDL-JobRequisition-Title|ZFRCE0571_ZBEN|ZFRCE001_ZBEN|
Brazil|ZBEN United Benefits Corporation|REQUISITION_DRAFT|REQUISITION_DELETED|ZFRCELOC1|US|CSP-DEFAULT|
DEFAULT_AF|HDL-JobRequisition-SysID-1|HRC_SQLLOADER
```

Prospect

Overview of Loading Prospects

You can load the prospect details comprising Prospect and Prospect Interaction using the HCM Data Loader (HDL).

Guidelines for Loading Prospect

A prospect can be of four types, namely Added, Agency Referral, Imported, and Referred.

This topic describes the considerations for loading prospects of the type "Added" using HCM Data Loader (HDL).

Prospect

User Key

The user key for prospect is RequisitionNumber, CandidateNumber.

ContextTypeCode

You can add prospects from various contexts such as Candidate Pool, Requisition, and Candidate Search by specifying the context type codes ORA_CANDIDATE_POOL, ORA_CANDIDATE_SEARCH and ORA_REQUISITION respectively.

ORA_REQUISITION

You should provide the requisition details AddedFromRequisitionId or AddedFromRequisitionNumber of the requisition from which the prospect is being added. The candidate should be an applicant or a prospect to the requisition from which it's getting added.

ORA_CANDIDATE_POOL

You should provide the AddedFromPoolId or AddedFromPoolName, and the AddedFromPoolStatus of the candidate pool from which the prospect is being added. The candidate should be a member of the candidate pool from which it's getting added.

Note: You can't add the same candidate as a prospect to a requisition from different contexts.

InactivateFlag

It's not possible to delete prospects. You can soft delete prospects by specifying InactivateFlag as 'Y'.

SendInviteFlag

- You may send emails inviting candidates to apply for jobs by setting the SendInviteFlag as Y. If you don't want emails to be sent to candidates, then set it to N.
- You can't add prospects to a linked requisition. A prospect can only be added to a standard or pipeline requisition that's posted, except for confidential job requisitions to which you can add prospects in open phase and not posted state.

AddedByPersonId

AddedByPersonId must be provided when creating a prospect. Adding a prospect to a job requisition must be performed by a job requisition team member.

Source Tracking

You can add source tracking information by providing the source and source medium. If you don't provide source information while creating the record, a default entry for source data will be created. Source tracking information can't be updated once created.

Rules for Requisition

- The following table lists the types of candidates and the requisition types they can apply for.

Requisition Type	Who can apply
Internal	Internal Candidates
External	External Candidates
Internal and external	Internal and External candidates
Confidential	Internal and External candidates

Requisition Type	Who can apply

- Prospect added to a requisition should not be canceled, filled or suspended.
- You can't add a candidate as a prospect to the requisition if it's already present as an active applicant on the requisition.

Prospect Interaction

A prospect interactions is always in the context of a job requisition. A job requisition team member adds a prospect interaction to a job requisition.

InteractionTypeCode

Interactions may consist of phone calls, emails, text messages, and in-person meetings.

AddedByPersonId

You must provide AddedByPersonId when you're creating a prospect interaction.

InteractionDate and InteractionType

You must provide InteractionDate and InteractionType when you create a prospect interaction. InteractionDate should be on or before today's date.

Examples of Loading Prospect

This topic provides examples that show how to load prospects.

Creating Prospect

This example Prospect.dat file creates a prospect record with the attributes ContextTypeCode, ProspectStatusCode, RequisitionNumber, CandidateNumber, AddedByPegrsonNumber, AddedFromPoolName, AddedFromPoolStatus and AddedFromRequisitionNumber.

```
METADATA|Prospect|ContextTypeCode|ProspectStatusCode|RequisitionNumber|CandidateNumber|AddedByPersonNumber|
AddedFromPoolName|AddedFromPoolStatus|AddedFromRequisitionNumber|SendInviteFlag|InactivateFlagMERGE|
Prospect|ORA_CANDIDATE_POOL|ORA_ACTIVE|ReqRB|263292|ZFRCE001_ZBEN|MU_Candidate_Pool_3|A|||MERGE|Prospect|
ORA_CANDIDATE_SEARCH|ORA_ACTIVE|ReqRB|262384|ZFRCE001_ZBEN|||||MERGE|Prospect|ORA_REQUISITION|ORA_ACTIVE|
ReqRB|263062|ZFRCE001_ZBEN|||DV-RQ-04||
```

This example Prospect.dat.file creates a prospect interaction record with the attributes InteractionDate, SourceSystemId, SourceSystemOwner, RequisitionNumber, CandidateNumber, AddedByPersonNumber, Text and InteractionTypeCode.

```
METADATA|ProspectInteraction|InteractionDate|SourceSystemId|SourceSystemOwner|RequisitionNumber|
CandidateNumber|AddedByPersonNumber|Text|InteractionTypeCodeMERGE|ProspectInteraction|2021/08/06 00:00:00|
ReqRB_263292_I1|HRC_SQLLOADER|ReqRB|263292|ZFRCE001_ZBEN|test3.txt|ORA_PHONEMERGE|ProspectInteraction|
2021/08/06 00:00:00|ReqRB_262384_I1|HRC_SQLLOADER|ReqRB|262384|ZFRCE001_ZBEN|test2.txt|ORA_EMAILMERGE|
ProspectInteraction|2021/08/06 00:00:00|ReqRB_263062_I1|HRC_SQLLOADER|ReqRB|263062|ZFRCE001_ZBEN|test1.txt|
ORA_TEXT
```

Adding Prospect Interaction to an existing Prospect

This example Prospect.dat file adds the prospect interaction details for a prospect using the RequisitionNumber, CandidateNumber and AddedByPersonNumber attributes.

```
METADATA|Prospect|ContextTypeCode|ProspectStatusCode|RequisitionNumber|CandidateNumber|AddedByPersonNumber|
AddedFromPoolName|AddedFromPoolStatus|AddedFromRequisitionNumber|SendInviteFlag|InactivateFlagMERGE|
Prospect|ORA_CANDIDATE_SEARCH|ORA_ACTIVE|ReqRB|262384|ZFRCE001_ZBEN|||||

METADATA|ProspectInteraction|InteractionDate|SourceSystemId|SourceSystemOwner|RequisitionNumber|
CandidateNumber|AddedByPersonNumber|Text|InteractionTypeCodeMERGE|ProspectInteraction|2021/08/06 00:00:00|
ReqRB_262384_I1|HRC_SQLLOADER|ReqRB|262384|ZFRCE001_ZBEN|test2.txt|ORA_EMAILMERGE|ProspectInteraction|
2021/08/06 00:00:00|ReqRB_262384_I2|HRC_SQLLOADER|ReqRB|262384|ZFRCE001_ZBEN|test1.txt|ORA_TEXT
```

Deleting Prospect

This example Prospect.dat file soft deletes the prospect record by setting the InactivateFlag to Y.

```
METADATA|Prospect|ContextTypeCode|ProspectStatusCode|RequisitionNumber|CandidateNumber|AddedByPersonNumber|
AddedFromPoolName|AddedFromPoolStatus|AddedFromRequisitionNumber|SendInviteFlag|InactivateFlagMERGE|
Prospect|ORA_CANDIDATE_POOL|ORA_ACTIVE|ReqRB|263292|ZFRCE001_ZBEN|MU_Candidate_Pool_3|A|||Y
```

Job Offers

Overview of Loading Job Offers

Using HCM Data Loader, you can create and update job offers for external and internal candidates who submitted a job application.

The Job Offers solution includes the following HCM Data Loader objects:

1. Job Offer
2. Worker
3. Assigned Payroll
4. Payroll Assignment Details
5. Salary
6. Element Entry
7. Job Offer Letter

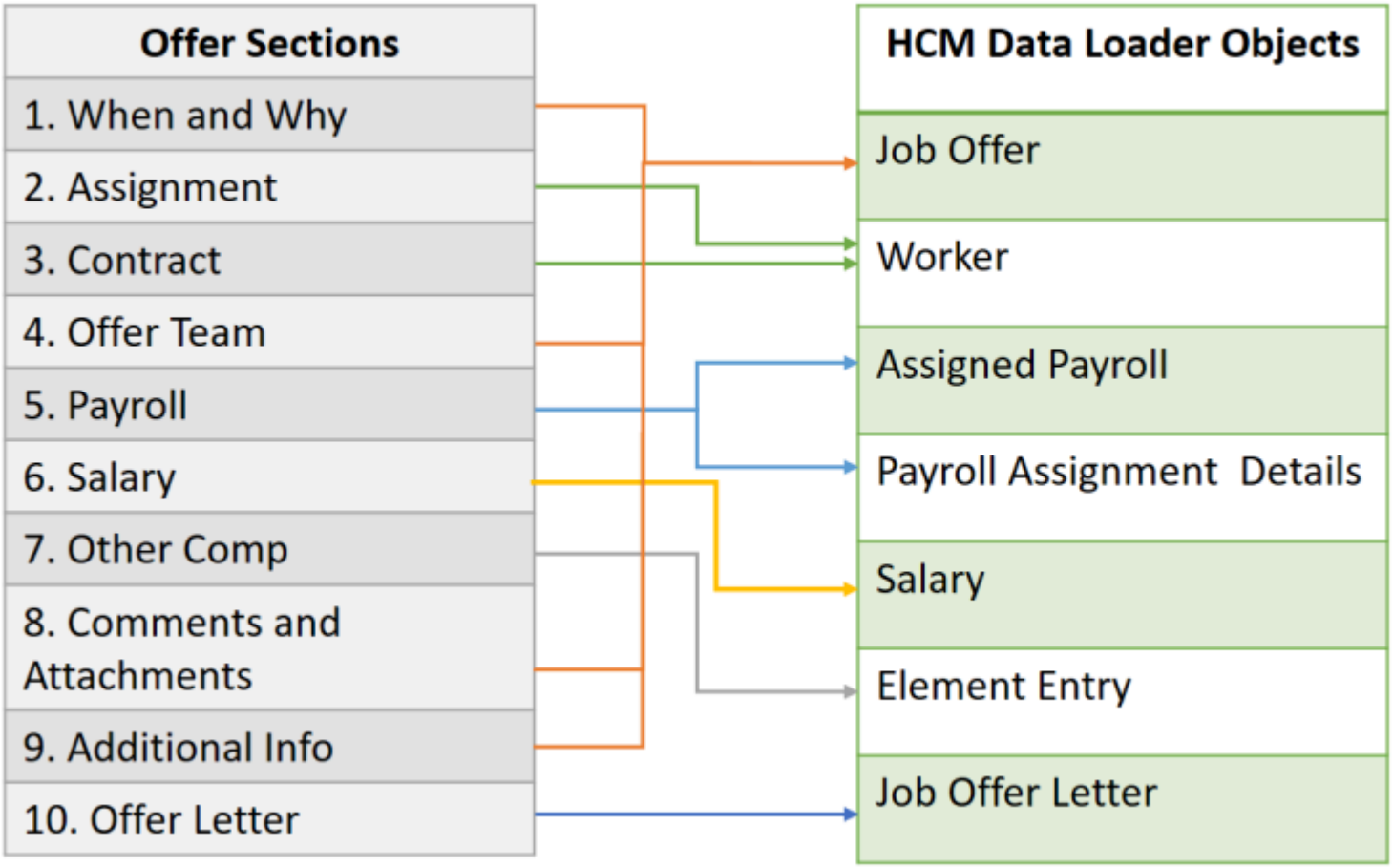
It's recommended that you add .dat files for these objects to a single zip file and upload them. HCM Data Loader will load them in the above order.

To submit files individually, submit them in the above order and ensure that the previous file completes loading before you submit the next file. This is because each file depends on the data uploaded by the previous files.

Offers in the Recruiting Application and HCM Data Loader

This figure shows how various sections of the Offer page in Recruiting correspond to the different objects in HCM Data Loader.

Each object contains information pertaining to one or more sections of the Offer page.



General Guidelines for Loading Job Offers

Consider these general guidelines when loading job offers using HCM Data Loader.

- To create or update an offer, the job application must be in the Job Offer phase and Draft state, or any other valid phase and state from where it can be moved to the Job Offer phase and To Be Created state.
- To create an offer, you must load the following objects:
  - Job Offer
  - Worker
  - Salary
  - Job Offer Letter
- It's recommended that you use source keys for identifying each record and for referencing the other objects in the Job Offers solution. For example, when you use a source key when defining the job offer assignment, use the same source key when referring to the assignment from the Salary, Payroll Assignment, Assigned Payroll, and Element Entry objects.

Note: You can't update the effective date for an offer-related assignment, payroll, salary, or compensation elements (element entries) using HCM Data Loader.

Object-Specific Guidelines for Loading Job Offers

Consider these guidelines for each object when loading job offers using HCM Data Loader.

Job Offer

- You can create and update a job offer only when the job application is in the Draft state.
- The value of the ProposedStartDate attribute must be the same as or later than the person start date and effective start date of the offer assignment during a job offer create or update.

Action Code Attribute

These values are predefined for the Action Code attribute:

Action Code	Action Name
ADD_PEN_WKR	Add Pending Worker
ORA_ADD_PWK_WORK_RELATION	Add Pending Work Relationship
GLB_TRANSFER	Global Transfer
GLB_TEMP_ASG	Global Temporary Assignment
HIRE_ADD_WORK_RELATION	Add Employee Work Relationship
ADD_CWK_WORK_RELATION	Add Contingent Work Relationship
ADD_ASSIGN	Add Assignment
TEMP_ASG	Temporary Assignment
ASG_CHANGE	Change Assignment
PROMOTION	Promotion
TRANSFER	Transfer
JOB_CHANGE	Job Change
LOCATION_CHANGE	Location Change
POSITION_CHANGE	Position Change
HIRE_ADD_WORK_RELATION	Add Employee Work Relationship

Action Code	Action Name
ADD_CWK_WORK_RELATION	Add Contingent Work Relationship

In addition to these, custom actions can also be created.

Job Offer Team

Job Offer Attachment

Attachments can be of these categories:

- MISC – Miscellaneous
- IRC_INTERNAL – Internal attachments with security applied. Document of Record (DoR) isn't available for this category. It's visible only in Oracle Recruiting if you have the Recruiter role.
- IRC_JOB_OFFER_CANDIDATE – Visible to both the candidate (DoR) and the recruiter.

Worker

You can create and update a job offer assignment using the Worker object.

- While it is required to include the Work Relationship, Work Terms, and Assignment child objects as part of the Worker.dat file, it's optional to include Work Measures (FTE) and Contract objects, as it depends on the business requirement.
- The following SET instruction should be placed at the top of the Worker.dat file used to load any job offer assignments:

```
SET OFFER_ASSIGNMENTS_FILE Y
```

> **Note:** This SET instruction will validate for offer-related information in the Worker.dat file. If you load record types that don't pertain to job offer assignments, your upload will fail.

- If a source key is used in the Worker.dat file to reference the job offer on the Assignment object, then it must have the same value as the source key for the job offer in the JobOffer.dat file.

Work Relationship

You can create and update work relationship details using this object.

- The PrimaryFlag attribute should be set to N.

Work Terms

Using this object, you can create and update the terms and conditions associated with one or more assignments in a work relationship.

- You must load the WorkTerms object for every create and update operation on the Assignment object.

Assignment

Using this object, you can create and update assignment details such as a worker's job, position, grade, department and location.

- You can't update the ProjectedStartDate attribute using the Assignment object. Instead, update it using the JobOffer.dat file.

Assignment Work Measures (FTE)

You can optionally create and update the Assignment Work Measures object.

- The Full Time Equivalent (FTE) record is automatically created by including this SET instruction at the top of the Worker.dat file:

```
SET CALCULATE_FTE Y
```

> **Note:** You must not include an AssignmentWorkMeasure record for the candidate when using this SET instruction.

Contract

You can create and update the contract details associated with an assignment or terms of employment.

- The Contract object must be provided only if the legal employer is configured to have a contract.

Payroll

These two objects, Assigned Payroll and Payroll Assignment Details, are relevant while loading job offers.

Assigned Payroll

Using this object, you can create, update, and delete the payroll frequency details for an assignment.

- You need to load this object if the salary is based on a payroll process. Otherwise, it's optional.
- You can't provide date-effective changes when loading assigned payroll data for job offers.

Payroll Assignment Details

Using this object, you can create or update payroll assignment details for an employee, such as time card required status and overtime period. However, you can't delete these details.

It's optional to load this object.

Salary

You can create, update, and delete proposed salary details for worker assignments.

- You must load the Salary object as part of the Job Offer load.
- Salary with rates isn't supported.
- You can't load date-effective changes for a job offer salary.

Element Entry

An element entry holds the actual payroll element and benefits granted to a worker. You can create, update, and delete element entries.

- Element entries represent the Individual Compensation Plans (ICPs) assigned in the offer. It maps to the Other Comp section in the offer. The number of element entries should be the same as the number of ICPs awarded to the candidate.
- It is optional to load the Element Entry object as part of the Job Offer load.
- The OtherCompElementCount attribute value in the JobOfferLetter.dat file must match the number of element entries in the offer.

Job Offer Letter

You can create and update offer letters using the Job Offer Letter object.

- You must load the Job Offer Letter object as part of the Job Offer load.
- You must specify an active offer letter template for submitting the offer and generating the offer letter.
  - The offer letter template must be defined for the same candidate type as the candidate mentioned in the job offer.
  - Offer letter templates are defined in the Recruiting Content Library.
- You must include a reference to the job offer for which the letter is intended.
- If the SubmitOffer attribute is set to Y, the offer gets submitted and the approval workflow is triggered. If it's set to N, the offer is created, but not submitted.

Related Topics

- Overview of Loading Workers
- Overview of Loading Payroll Details
- Guidelines for Loading Salary Records
- Overview of Loading Element Entries

Examples of Loading Job Offers

Here are sample files that show how to create a job offer using HCM Data Loader.

Example for an External Candidate

You can use the sample JobOffer.dat file provided here to create or update a job offer for external candidates.

Job Offer

This example creates an offer for the candidate CAN614005. The source key VISION | CAN614005-R23522 uniquely identifies the job offer. This source key value is used to identify the team members and the attachment for the job offer.

```
METADATA|JobOffer|SourceSystemOwner|SourceSystemId|CandidateNumber|RequisitionNumber|Comments|
OfferComments|Additionaltext1|Additionaltext2|HiringManagerPersonNumber|HiringManagerAssignmentNumber|
RecruiterPersonNumber|RecruiterAssignmentNumber|ProposedStartDate|LegalEmployer|WorkerType|ActionCode|
ObjectStatus|FLEX:IRC_OFFERS_DFF|OfferText(IRC_OFFERS_DFF=FxF_Offer_Context)
```

```
MERGE|JobOffer|VISION|CAN614005-R23522|CAN614005|R23522|Comments for the job offer|CAN614005-
R23522_OfferComments.txt|CAN614005-R23522_Additionaltext1.txt|CAN614005-R23522_Additionaltext2.txt|431509|
E431509|451255|E451255|2022/10/03|VISION US|Employee|ADD_PEN_WKR||VISION_Offer_Context|Custom descriptive
 flexfield example
METADATA|JobOfferTeam|SourceSystemOwner|SourceSystemId|OfferId(SourceSystemId)|CollaboratorType|MemberType|
PersonNumber|AssignmentNumber
MERGE|JobOfferTeam|VISION|CAN614005-R23522-E411302|CAN614005-R23522||ORA_COLLABORATOR|411302|E411302
MERGE|JobOfferTeam|VISION|CAN614005-R23522-E396684|CAN614005-R23522|ORA_HIRING_TEAM_COLLABORATOR|
ORA_COLLABORATOR|396684|E396684
METADATA|JobOfferAttachment|OfferId(SourceSystemOwner)|Title|CandidateNumber|RequisitionNumber|
OfferId(SourceSystemId)|File|DataTypeCode|Category|URLorFileName
MERGE|JobOfferAttachment|VISION|CAN614005-Resume-File|CAN614005|R23522|CAN614005-R23522|
CAN614005_Resume.docx|FILE|IRC_INTERNAL|CAN614005_Resume.docx
```

To load data into a Character Large Object (CLOB) object, name the file that contains the data to be loaded and place that file in a subfolder of the same compressed zip file used to load the JobOffer.dat file. The subfolder must be named ClobFiles.

The values for these attributes need to be provided in separate files within the ClobFiles subfolder:

- OfferComments
- Additionaltext1
- Additionaltext2

Offer Assignment

Use the Worker.dat file to create the offer assignment.

You must always include the `SET OFFER_ASSIGNMENTS_F Y` instruction at the top of the file. Optionally, include the `SET CALCULATE_FTE Y` instruction if you want the full-time equivalent to be created.

This example creates the work relationship, employment (work) terms and assignment for the candidate referenced in the Job Offer example provided earlier.

The job offer is referenced on the Assignment record using the OfferId(SourceSystemId) attribute. The source system ID value CAN614005-R23522 is the same as supplied to the SourceSystemId attribute on the JobOffer record in the previous example.

```
SET OFFER_ASSIGNMENTS_FILE Y
SET CALCULATE_FTE Y
METADATA|WorkRelationship|SourceSystemOwner|SourceSystemId|PersonId|DateStart|WorkerType|LegalEmployerName|
PrimaryFlag|ActionCode
MERGE|WorkRelationship|VISION|WRCAN614005|300100574088451|2022/10/03|O|VISION US|N|EMPL_OFFER_CREATE
METADATA|WorkTerms|SourceSystemOwner|SourceSystemId|PeriodOfServiceId(SourceSystemId)|EffectiveStartDate|
EffectiveSequence|EffectiveLatestChange|SystemPersonType|ActionCode|AssignmentType|BusinessUnitShortCode|
GradeLadderPgmName|PersonTypeCode|AssignmentNumber
MERGE|WorkTerms|VISION|WTCAN614005|WRCAN614005|2022/10/03|1|Y|ORA_CANDIDATE|EMPL_OFFER_CREATE|OT|Vision
 Business Unit|CMP_SBwithGLST|Recruiting Candidate|
METADATA|Assignment|SourceSystemOwner|SourceSystemId|WorkTermsAssignmentId(SourceSystemId)|
EffectiveStartDate|EffectiveSequence|EffectiveLatestChange|OfferId(SourceSystemId)|ActionCode|
AssignmentType|BusinessUnitShortCode|GradeLadderPgmName|PersonTypeCode|AssignmentNumber|
NormalHours|Frequency|PositionCode|JobCode|GradeCode|LocationCode|DepartmentName|ProjectedStartDate|
ProposedUserPersonType|SystemPersonType|TaxReportingUnit
MERGE|Assignment|VISION|ECAN614005|WTCAN614005|2022/10/03|1|Y|CAN614005-R23522|EMPL_OFFER_CREATE|O|Vision
 Business Unit|CMP_SBwithGLST|Recruiting Candidate||30|W|Analyst Position|Analyst|IC3|VUSSF_01|US HCM|
2022/10/03|Employee|ORA_CANDIDATE|VIS USA
```

Payroll Assignment Details

Use the PayrollAssignmentDetails.dat file to create and maintain the payroll assignment details, such as time card required status and overtime period.

The offer assignment for which the payroll assignment details are being created is referenced by its source key using the AssignmentId(SourceSystemId) attribute and AssignmentId(SourceSystemOwner) attributes.

The source key value ECAN614005 | VISION that you defined when creating the offer assignment is used in this example.

```
METADATA|PayrollAssignmentDetails|AssignmentId(SourceSystemOwner)|AssignmentId(SourceSystemId)|
EffectiveStartDate|LegislativeDataGroupName|TimecardRequiredFlag|OvertimePeriodCode
MERGE|PayrollAssignmentDetails|VISION|ECAN614005|2022/10/03|United States LDG|Y|VIS_WK
```

Assigned Payroll

Use the AssignedPayroll.dat file to create and maintain payroll assignment details, such as the payroll name, frequency details, and related start and end dates.

The offer assignment for which the assigned payroll details are being created is referenced by its source key using the attributes AssignmentId(SourceSystemId) and SourceSystemOwner. The source key value ECAN614005 | VISION that you defined when creating the offer assignment is used in this example.

```
METADATA|AssignedPayroll|SourceSystemOwner|SourceSystemId|AssignmentId(SourceSystemId)|EffectiveStartDate|
PayrollDefinitionCode|LegislativeDataGroupName|StartDate|OvertimePeriodCode|TimecardRequiredFlag
MERGE|AssignedPayroll|VISION|ECAN614005_AP|ECAN614005|2022/10/03|VIS_MONTHLY_PAYROLL|United States LDG|
2022/10/03|VIS_WK|Y
```

Salary

Use the Salary.dat file to create or update salary proposals.

The offer assignment for which salary details are being created is referenced by its source key using the attributes AssignmentId(SourceSystemId) and SourceSystemOwner. The source key value ECAN614005 | VISION that you defined when creating the offer assignment is used in this example.

This example uses salary components.

```
METADATA|Salary|SourceSystemOwner|SourceSystemId|SalaryBasisName|AssignmentId(SourceSystemId)|DateFrom|
SalaryAmount|MultipleComponents|NextSalReviewDate|SalaryApproved|ActionCode
MERGE|Salary|VISION|ECAN614005_1|STD_CMP_AUTO_BUZoneGR_001|ECAN614005|2022/10/03|0|Y|2022/11/01|Y|
EMPL_OFFER_CREATE
METADATA|SalarySimpleComponent|SourceSystemOwner|SourceSystemId|SalaryId(SourceSystemId)|
AssignmentId(SourceSystemId)|SalaryDateFrom|ComponentCode|Amount|Percentage
MERGE|SalarySimpleComponent|VISION|ECAN614005_1_BAS|ECAN614005_1|ECAN614005|2022/10/03|ORA_BASIC|60000|
MERGE|SalarySimpleComponent|VISION|ECAN614005_1_CTY|ECAN614005_1|ECAN614005|2022/10/03|ORA_CITY_ALLOWANCE||
10
MERGE|SalarySimpleComponent|VISION|ECAN614005_1_TRN|ECAN614005_1|ECAN614005|2022/10/03|
ORA_TRANSPORT_ALLOWANCE|2500|
MERGE|SalarySimpleComponent|VISION|ECAN614005_1_TVL|ECAN614005_1|ECAN614005|2022/10/03|
ORA_TRAVEL_ALLOWANCE||8
MERGE|SalarySimpleComponent|VISION|ECAN614005_1_SPC|ECAN614005_1|ECAN614005|2022/10/03|
ORA_SPECIAL_ALLOWANCE||100
MERGE|SalarySimpleComponent|VISION|ECAN614005_1_HOU|ECAN614005_1|ECAN614005|2022/10/03|
ORA_HOUSING_ALLOWANCE|10000|
MERGE|SalarySimpleComponent|VISION|ECAN614005_1_SAL|ECAN614005_1|ECAN614005|2022/10/03|ORA_OVERALL_SALARY||
```

Element Entry

Use the ElementEntry.dat file to create or maintain compensation elements for the candidate.

The offer assignment for the element entries is referenced by its source key using the HrAssignmentId(SourceSystemId) and SourceSystemOwner attributes. The source key value ECAN614005 | VISION that you defined when creating the offer assignment is used in this example.

This example creates three element entries:

Element Entry	Amount
SpotBonus	1000.325
HiringBonus	5467.25
INF_ISO	100 shares

```
METADATA|ElementEntry|SourceSystemOwner|SourceSystemId|HrAssignmentId(SourceSystemId)|EffectiveStartDate|
EffectiveEndDate|ElementName|LegislativeDataGroupName|EntryType|MultipleEntryCount|CreatorType
MERGE|ElementEntry|VISION|ECAN614005_SB|ECAN614005|2022/10/03|2022/10/30|SpotBonus|United States LDG|E||VC
MERGE|ElementEntry|VISION|ECAN614005_HB|ECAN614005|2022/10/03||HiringBonus|United States LDG|E|1|VC
MERGE|ElementEntry|VISION|ECAN614005_STK|ECAN614005|2022/10/03||INF_ISO|United States LDG|E|1|VC
METADATA|ElementEntryValue|SourceSystemOwner|SourceSystemId|ElementEntryId(SourceSystemId)|
EffectiveStartDate|EffectiveEndDate|ElementName|LegislativeDataGroupName|EntryType|MultipleEntryCount|
InputValueName|ScreenEntryValue
MERGE|ElementEntryValue|VISION|ECAN614005_SB_AMT|ECAN614005_SB|2022/10/03|2022/10/30|SpotBonus|United States
 LDG|E||Amount|1000.325
MERGE|ElementEntryValue|VISION|ECAN614005_HB_AMT|ECAN614005_HB|2022/10/03||HiringBonus|United States LDG|E|
1|Amount|5467.25
MERGE|ElementEntryValue|VISION|ECAN614005_STK_GRNT|ECAN614005_STK|2022/10/03||INF_ISO|United States LDG|E|1|
Shares Granted|100
```

Job Offer Letter

Use the JobOfferLetter.dat file to create or update a job offer letter.

Specify a value of `Y` in the HasPayroll attribute if you want to validate that the candidate has a payroll record. Specify a number in the OtherCompElementCount attribute to validate the total number of compensation elements expected in the offer.

The SubmitOffer attribute is used to submit the job offer for approval. When the job offer is submitted, it moves to the Approved or Pending Approval status depending on whether Bypass Approval is enabled.

```
METADATA|JobOfferLetter|CandidateNumber|RequisitionNumber|OfferLetterContentItemCode|HasPayroll|
OtherCompElementCount|SubmitOffer|ExpirationDate
MERGE|JobOfferLetter|CAN614005|R23522|OR_28SEP2022_2009PM|N|3|Y|2022/10/12 21:00:00
```

Example for an Internal Candidate

You can use the sample JobOffer.dat file provided here to create or update a job offer for internal candidates.

For the other objects, you can refer to the sample files provided earlier for external candidates, and use data that's relevant for internal candidates.

Job Offer

This example creates an offer for an internal candidate 283706. The source key VISION | 283706_R712413 uniquely identifies the job offer. This source key value is used to identify the team members and attachment for the job offer.

This example differs from the external candidate job offer in the value for the ActionCode. Here the action is global transfer (GLB_TRANSFER).

```
METADATA|JobOffer|SourceSystemOwner|SourceSystemId|CandidateNumber|RequisitionNumber|Comments|
OfferComments|Additionaltext1|Additionaltext2|HiringManagerPersonNumber|HiringManagerAssignmentNumber|
RecruiterPersonNumber|RecruiterAssignmentNumber|ProposedStartDate|LegalEmployer|WorkerType|ActionCode
MERGE|JobOffer|VISION|283706_R712413|283706|R712413|Global Transfer US-UK, GT-HDLINT, OFF-Virgillo
 Past Date|283706_R712413 OfferComments.txt|283706_R712413 Additionaltext1.txt|283706_R712413
 Additionaltext2.txt|131509|E131509|132552|E132552|2022/11/06|VISION US|Employee|GLB_TRANSFER
METADATA|JobOfferTeam|SourceSystemOwner|SourceSystemId|OfferId(SourceSystemId)|CollaboratorType|MemberType|
PersonNumber|AssignmentNumber
MERGE|JobOfferTeam|VISION|283706_R712413_E121302|283706_R712413|IRC_INTERVIEW_TEAM|ORA_COLLABORATOR|121302|
E121302
MERGE|JobOfferTeam|VISION|283706_R712413_E211337|283706_R712413|IRC_REC_COORDINATOR|ORA_COLLABORATOR|211337|
E211337
MERGE|JobOfferTeam|VISION|283706_R712413_E231502|283706_R712413|ORA_HIRING_TEAM_COLLABORATOR|
ORA_COLLABORATOR|231502|E231502
MERGE|JobOfferTeam|VISION|283706_R712413_E233001|283706_R712413||ORA_COLLABORATOR|233001|E233001
METADATA|JobOfferAttachment|OfferId(SourceSystemOwner)|OfferId(SourceSystemId)|CandidateNumber|
RequisitionNumber|Title|File|DataTypeCode|URLorFileName|Category
MERGE|JobOfferAttachment|VISION|283706_R712413|283706|R712413|Resume-File1|283706 Resume1.docx|FILE|
Resume1.docx|IRC_JOB_OFFER_CANDIDATE
MERGE|JobOfferAttachment|VISION|283706_R712413|283706|R712413|Resume-File2|283706 Resume2.docx|FILE|
Resume2.docx|IRC_INTERNAL
MERGE|JobOfferAttachment|VISION|283706_R712413|283706|R712413|Resume-Link1||WEB_PAGE|http://www.google.com|
MISC
```