

# Oracle Fusion Cloud HCM

---

## **HCM Data Loader Questions and Answers**

Oracle Fusion Cloud HCM  
HCM Data Loader Questions and Answers

G37236-02

*Copyright* © 2025, Oracle and/or its affiliates.

Author: Oracle Human Resources Information Development Team

# Contents

## Get Help

---

i

<b>1</b>	<b>HCM Data Loader Questions and Answers</b>	<b>1</b>
	What's HCM Data Loader?	1
	When should I use HDL?	1
	How does HCM Data Loader (HDL) process data during a load?	1
	How do I control which business objects users can load with HCM Data Loader?	2
	What key types does HDL support, and when should I use each?	3
	Where can I review the complete list of supported business objects?	4
	What are source keys in HCM Data Loader?	5
	What are user keys in HCM Data Loader?	6
	When do I use user keys?	6
	What are Oracle Fusion GUIDs in HCM Data Loader?	6
	What are Oracle Fusion surrogate IDs in HCM Data Loader?	7
	What's meant by business object structure?	7
	What type of files can be loaded into the HDL?	9
	What are the various file line instructions used in an HDL file?	10
	What are file discriminators?	11
	How to configure METADATA instruction?	12
	How to configure MERGE instruction?	13
	How to configure DELETE instruction?	14
	How to configure SET instruction?	14
	How do I generate a single template file?	19
	How do I generate template files for all business objects?	20
	How do I use template files?	20
	How do I view the structure, supported key systems, data formats, and required attributes for business objects?	20
	How can I set up the configuration parameters for HCM Data Loader and HCM Spreadsheet Data Loader import and load processes?	22
	What are the various HCM Data Loader configuration parameters?	23
	How do I audit the changes done to HCM Data Loader parameters?	29
	How do I define referenced objects?	30

How do I review list of values business object attributes?	31
How do I define the source-system owner values?	31
How can I test the HCM Data Loader process flow and connections?	32
How do I enable and manage audits for bulk-loaded objects?	33
How do I prepare data for loading successfully into Oracle HCM Cloud?	35
How do I load images, attachments, and large strings?	36
How do I include source keys in my data file?	37
How do I update source key associated with any record?	38
How do I include user keys in my data file?	39
How do I include Oracle Fusion Surrogate IDs in my data file?	40
How do I include Oracle Fusion GUIDs in my data file?	40
How do I include reserved characters in my data file?	41
How do I load and maintain translated business objects?	42
How do I include reference values in data files?	43
How do I delete a business object or its components?	45
What are the considerations for loading date-effective objects?	46
How do I load objects with multiple changes per day?	47
How are updates to future-dated records processed in Retain and Replace maintenance modes?	49
How do I set the Retain and Replace maintenance modes?	50
How are future-dated records preserved when using Retain mode?	51
What are the options to carry forward changed attribute values in Retain mode?	52
How are future-dated records handled in Replace mode when you replace attribute values?	54
How do I change the first effective start date or last effective end date for some business objects?	55
What are the guidelines for loading descriptive and extensible flexfield data?	56
How is data loaded to Oracle HCM Cloud?	58
How to extend the period of retention for a data set?	59
How processed data sets are deleted from stage tables?	59
How Delete HCM Data Loader Stage Table Data process deletes data sets?	60
How Delete HCM Spreadsheet Data Loader Stage Table Data process deletes spreadsheet data sets?	60
What happens to the archived data sets?	61
How automatic maintenance of the stage tables works in conjunction with a user request to delete stage table data?	61
How do I maintain the stage tables for HCM Data Loader?	62
How do I maintain the stage tables for HCM Spreadsheet Data Loader?	65
What are the recommended HCM Data Loader best practices?	68
What are the folders do I use to create your reports within OTBI and BI Publisher?	69
How do I change the first effective start date or last effective end date for some business objects?	70

What are the guidelines for loading descriptive and extensible flexfield data?	71
How do I include descriptive flexfield data using METADATA and MERGE lines in a .dat file?	73
How do I include extensible flexfield data using METADATA and MERGE lines in a .dat file?	75
How do I set up encryption for file transfer?	77
What are the keys used for encryption, decryption, signing, and verification in each supported encryption mode?	78
How do I import a public key for encrypting outbound files?	78
How do I generate a PGP encryption key pair?	79
Why must I use the alias fusion-key when generating the PGP key pair?	79
What's the role of the passphrase when creating a key pair?	80
How do I encrypt and upload files automatically?	80
What are encryption algorithms are supported by Oracle HCM Cloud?	80
How do I encrypt files in Microsoft Windows and Linux environments?	80
How do I load encrypted files to Oracle HCM Cloud from the Oracle WebCenter Content server?	81



# 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

Some application pages have help icons  to give you access to contextual help. 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. If the page has contextual help, help icons will appear.

## 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.

## 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](mailto:oracle_fusion_applications_help_ww_grp@oracle.com).

Thanks for helping us improve our user assistance!



# 1 HCM Data Loader Questions and Answers

## What's HCM Data Loader?

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.

You can use HCM Data Loader for data migration, ongoing maintenance of HCM data, and coexistence scenarios, where core HR data is uploaded regularly.

## When should I use HDL?

Use HDL to bulk-load or update complex HCM data into Oracle HCM Cloud efficiently. Typical use cases are as follows:

**Full implementations:** You perform a one-time migration of existing data in Oracle HCM Cloud and manage it thereafter.

**Coexistence implementation:** Oracle HCM Cloud can be used alongside other source systems and use HDL to periodically load data like person records, employment changes, compensation updates, or performance feedback.

You can load current or historical data, perform mass corrections to update it.

HCM Data Loader supports the loading of multi-level business objects with parent-child relationships.

## How does HCM Data Loader (HDL) process data during a load?

HDL follows a two phase process:

### Import Phase:

- Place a .zip file containing your .dat files on the Oracle WebCenter Content server. You can upload your file directly to the server, or use the HCM Data Loader interface for local files.
- Submit a request to HCM Data Loader to import and load the .zip file. For this step, you can use either the HCM Data Loader interface or the HcmCommonDataLoader web service.
- HCM Data Loader decompresses the .zip file and imports individual data lines to its stage tables. In the stage tables, related data lines are grouped to form business objects.
  - Any errors that occur during the import phase are reported on the HCM Data Loader interface.

### Load Phase:

- HCM Data Loader elevates the user before calling the relevant logical object interface method (delivered in product services) to load valid objects to the application tables.

- Any errors that occur during the load phase are reported in the HCM Data Loader interface.
- Review any errors from the import and load phases. You can perform this step either on the HCM Data Loader interface or using the HCM Data Loader Data Set Summary extract.
- Correct errors from the import and load phases in your source data.
- Load a new .zip file containing the corrected data to the WebCenter Content server.

Initiate import and load request until all of the data is successfully loaded.

**Note:** When you load data using the elevated user, HCM Data Loader bypasses the data security policies that normally apply to the session user. However, you must provide the data in the language configured for the `FUSION_APPS_HCM_ESS_LOADER_APPID` user - typically in the base language of your environment.

## How do I control which business objects users can load with HCM Data Loader?

You can manage access in three steps:

First, create custom user role to access **Configure HCM Data Loader Task**

Second, enable the capability of configuring the role based access of the business objects - Under **Setup and Maintenance > HCM Data Loader > Configure HCM Data Loader**, set the **Enable Configuration of Role-Based Business Object Access parameter** to **Yes**.

**Note:** Data sets imported prior to enabling this feature can continue to be processed regardless of the user's business object access. Business object access is only evaluated when importing a data set.

Third, assign access by role or business object -

By Role:

- ◦ Access the **HCM Data Loader Business Object Access** task, select a role, and use the **Assign** drop-down to grant access to -
  - Individual business objects -
    - Search and select the business objects in the **Search and Select Business Objects** dialog box.
    - Click **Add** to add the selected business objects to the role. An entry appears in the **Assigned Business Objects** section for each of the selected business object.
  - All business objects within a product area -
    - Select the product area in the **Select Product Area** dialog box.
    - Click **Add**. A single entry appears for the product area in the **Assigned Business Objects** section.

**Note:** This role will have access to load data for all the business objects within the product area. Use the **View Business Objects** task to see which business objects are available in each product area.

- All unrestricted business objects -
  - o A warning message appears to indicate that users with this role can bulk-load data with any business object that doesn't load security-related data.
  - o Click **Add** to close the warning and continue. A single entry appears for all unrestricted business objects in the **Assigned Business Objects** section.
- All objects, including security-related objects -
  - o A warning message appears to indicate that users with this role will be able to use the security-related objects only if they have the **Load HCM Security Data** function security privilege
  - o Click **Add** to close the warning and continue. A single entry appears for all business objects for all product areas in the **Assigned Business Objects** section.

**Note:** In **Configure HCM Data Loader**, when **Restrict Access to Security Related Business Objects** is set to **Yes**, then only users with custom role with **Load HCM Security Data** function security privilege will be able to bulk-load data using security-related business objects.

By Business Object:

- Access the **HCM Data Loader Business Object Access** task, select **By Business Object** option, and use enter the name of the business object in the **Business Object** field. In the results section, select the business object -
  - o In the **Assigned Roles** section, click **Add**.
  - o Select one or more roles using **Search and select** .
  - o Click **Add** to see an entry in the **Assigned Roles** section for each selected role.

## What key types does HDL support, and when should I use each?

HDL supports **Oracle Fusion Globally Unique Identifier (GUID)**, **Source key**, **Oracle Fusion Surrogate ID**, and **User key** to uniquely identify records.

Unique keys identify:

- Record that you're creating or updating
- Parent of the record that you're creating or updating
- Parent record that may either be in the same file or exist already in Oracle HCM Cloud
- Any object referenced by the record you're creating or updating

If multiple keys are supplied, HDL prioritizes GUID, then Source Key, then Surrogate ID, then User Key.

HDL Key Type	Create/Update	Held on Object	Key Format	Auto-generated
GUID	Update Only	No	Hexadecimal	Yes

HDL Key Type	Create/Update	Held on Object	Key Format	Auto-generated
Surrogate ID	Update (See Note)	Yes	Numeric	Yes
Source Key	Create and Update	No	Alphanumeric	Conditionally (see Note)
User Key	Create and Update (see Note)	Yes	Alphanumeric	No

**Note:** You can use surrogate IDs to update objects, but these IDs may not be easily accessible to most Oracle HCM Cloud users. Source keys are automatically generated only if you don't provide them during object creation. User keys alone may not be valid for updating certain objects, as their values can change over time and may not uniquely identify the latest record. Keys that aren't held on the object exist in the Integration Key Map table.

**Foreign Object References:**

Oracle Fusion GUIDs and surrogate IDs are generated only after the corresponding record has been successfully created in Oracle HCM Cloud. Similarly, source keys are not recognized until the associated record exists in the system.

Therefore, when loading data that references **foreign objects** (i.e., other business objects), you must first ensure those referenced objects already exist in Oracle HCM Cloud.

For new implementations, it's recommended to load each business object individually and confirm a successful load before proceeding to the next.

If you include multiple business objects in a single `.zip` file, HCM Data Loader will load them based on object dependency order. In this case, a reference to a foreign object will fail only if the referenced object itself failed to load.

## Where can I review the complete list of supported business objects?

You can review the complete list of supported business objects using the **View Business Objects** task in the **Data Exchange** work area.

You can load business objects for most Oracle Fusion Human Capital Management (HCM) products like Oracle Fusion Absence Management, Compensation, Global Human Resources, Global Payroll, Performance Management, Profile Management, Talent Review and Succession Management, and Workforce Management. You can also update business objects using HCM Data Loader, regardless of how they were created.

## What are source keys in HCM Data Loader?

A source key are unique identifiers used by HCM Data Loader to distinguish business objects records within their source system during creation or update. Source keys are supported only for integration-enabled business objects.

It's recommended to use source keys for all implementations particularly for ongoing integrations, where data is maintained in the source system.

**Source-Key Structure:** A source key has two components -

Component	Description
Source-system owner	Identifies the originating system of the data.
Source-system ID	ID that uniquely identifies the business-object component and source-system owner in the source system.

Source-system owner component in the key allows you to load data from multiple source systems for the same business object.

For example, you may have person data in both US and UK databases to combine in one Oracle HCM Cloud system. The source-system ID doesn't have to be unique across both source systems. It must be unique only for the business-object component and source-system owner.

If you don't have a clear source-system ID for a record, then you can generate or derive one. For example, you could derive a source-system ID for a person address by concatenating the person number with the address type.

These source keys aren't stored directly on the created record—they are maintained separately in the **Integration Key Map** table.

**Source Keys for Date-Effective Business Objects:** When loading date-effective history for a business object, you must provide a source-system ID for every date-effective record in the file. The source-system ID must remain the same across all effective-dated rows for that object to ensure they are treated as part of the same record history.

**Default Source keys:** If you don't supply a source key when creating an object, HDL automatically generates a default source key.

- The default source-system owner is `FUSION`
- The default source-system ID is the surrogate ID of the object

You can later update or assign a new source key using the Source Key business object. To do this, provide the new source-system owner and source-system ID, along with a reference to the object you want to update.

**Source-System Owner Values:** The SourceSystemOwner value is validated against the `HRC_SOURCE_SYSTEM_OWNER` lookup type. Before loading data that uses custom source-system owners, you must add your external system names to this lookup. To update the lookup: Go to **Setup and Maintenance**, Use the **Manage Common Lookups** task, Search for and update `HRC_SOURCE_SYSTEM_OWNER` lookup type.

## What are user keys in HCM Data Loader?

Most Oracle HCM Cloud business objects, regardless of whether they're enabled for integration, have one or more attributes that make up a user key. The user key, which is also known as a natural key, is always visible on the user interface.

For example:

- The user key for an organization is the organization name.
- The user key for an area of responsibility is the responsibility name plus the person number.

## When do I use user keys?

User keys are part of the business object definition. They're always required when you create a logical object, regardless of how you create it.

You can also use user keys when updating logical objects. However, the value of a user key can change, and some user-key attributes are translatable. For this reason, you can't update some business objects if you supply only user keys to reference them. Therefore, whenever possible, you're recommended to use source keys when updating objects.

User keys are recommended when:

- You're referencing or maintaining an object that wasn't created with a source key.
- The source-key value is unknown.

### User Keys for Child Objects

When a business object is bound by another, the user key must include the user key for its parent. For example:

- Jobs are always part of a set. Therefore, job code alone doesn't uniquely identify a job. Instead, the set code must be part of the user key for a job.
- Job grades apply to a specific job. Therefore, the user key for a job grade must include both the user key for the grade and the user key for the parent job. Therefore, the user key for the job grade is made up of the grade code, job code, and set code.

## What are Oracle Fusion GUIDs in HCM Data Loader?

Oracle Fusion Globally Unique Identifiers (GUIDs) is one of the four types of keys that HCM Data Loader supports.

### Oracle Fusion GUIDs

When you create an integration-enabled business object in Oracle HCM Cloud, a GUID is generated for the object automatically. As the GUID is generated when you create the business object, it's suitable only for identifying business objects that already exist. You can't supply a GUID when creating a business object. The GUID, a hexadecimal value, isn't held on the created object but exists in the Integration Key Map table.

GUIDs are useful when you share changes with down-stream applications, such as third-party payroll providers. You extract the changes for the third-party payroll and provide the GUID as the unique key to the worker. When reporting any updates back to Oracle HCM Cloud, you can use GUIDs to identify the records to update.

## What are Oracle Fusion surrogate IDs in HCM Data Loader?

Oracle Fusion Surrogate IDs is one of the four types of keys that HCM Data Loader supports.

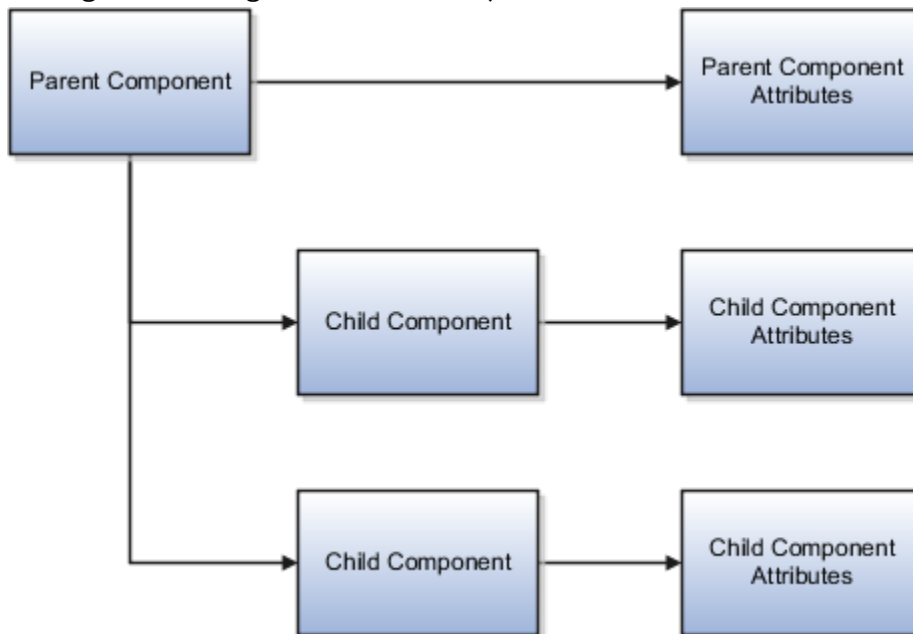
### Oracle Fusion Surrogate IDs

The surrogate ID, a numeric value, is the internal system identifier that's generated when you create a business object. All objects, including those that aren't integration-enabled, have surrogate IDs. The surrogate ID is held on the object. The ID is also used in the Integration Key Map table to provide a mapping for GUID and source-key values. As the surrogate ID isn't generated until the object is created, you can't use this value when creating objects. Access to surrogate IDs may also be limited. On-premises customers are more likely to have access to surrogate IDs than Oracle HCM Cloud customers.

## What's meant by business object structure?

Each Oracle HCM Cloud business object is a hierarchy of business-object components. At the top of the hierarchy is the parent component, and beneath it may be child and grandchild components. Each business-object component has multiple attributes.

This figure shows a generic business-object structure.



For example, the Rating Model business object includes Rating Model, Rating Level, and Rating Category components. The Rating Model component is the parent of the other two components. Each component has attributes such as **RatingName**, **RatingModelCode**, **RatingLevelCode**, and so on.

The most complex business object supported by HCM Data Loader is the Worker object, where five levels exist in the object hierarchy. These range from the Worker component at the top to Assignment Work Measure, Assignment Manager, Assignment Grade Steps, and Assignment Extra Information at the bottom. By contrast, the Person Type object has only the Person Type component.

## Terminology

This table defines the terms used by HCM Data Loader to refer to business objects.

Term	Meaning
Object or business object	Refers to the complete object, which is the parent component and all child components. For example, Grade and Worker are business objects.
Component or business-object component	Refers to components of a business object. For example, Person Name and Work Relationship are components of the Worker object.
Logical object	Refers to a group of related components that form one occurrence of a business object. For example, the grade IC1 is a logical object.

## Loading Logical Objects

When you deliver multiple components for the same business object together, HCM Data Loader groups them and loads the complete logical object. It doesn't process the components separately. If any component of the logical object fails validation, then the whole logical object is rejected. As HCM Data Loader loads only complete logical objects, you can be sure of exactly which data has been loaded. For example, when loading jobs, you know which jobs loaded successfully and which jobs failed to load.

## Integration-Enabled Objects

HCM Data Loader can load any integration-enabled business object. An integration-enabled business object supports these four key types:

- Oracle Fusion GUID
- Oracle Fusion surrogate ID
- Source key
- User key

An integration-enabled object has an entry in the Integration Key Map table, where its GUIDs and source keys exist.

HCM Data Loader can also load a few objects, such as Department Tree, that aren't integration enabled. Typically, Oracle HCM Cloud doesn't own such objects and although you can't identify the object by a source key nor GUID, you can often reference foreign objects using integration keys, such as the parent and child departments while creating a department tree.

**Tip:** You can review business-object details on the View Business Objects page. The **Integration Keys Supported** field on the Component Details tab for a selected component indicates whether the component is integration enabled.

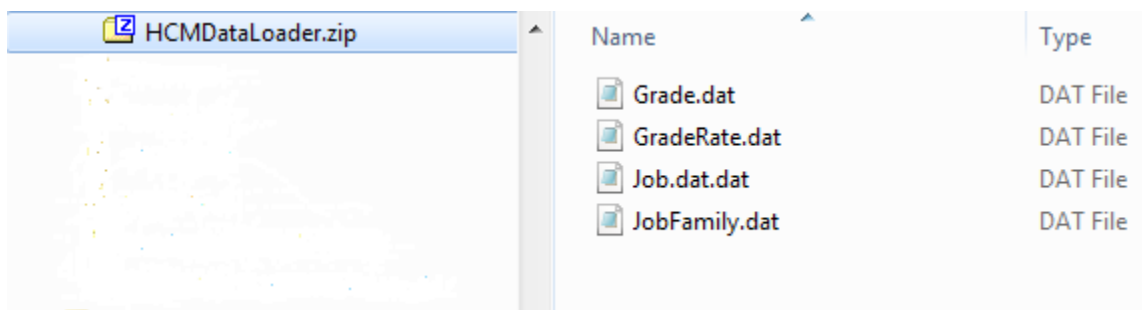
Here's a tutorial on using the View Business Objects task which is the data dictionary for all supported business objects:  
[View Business Object task](#)

## What type of files can be loaded into the HDL?

You use .zip files to load data with HCM Data Loader. Each .zip file can contain one or more data (.dat) file. Each file includes data for a single business object, such as Location, Grade, Worker, or Salary.

Your data file has the name of a business object that's specified by HCM Data Loader, such as Job.dat or Worker.dat. In any .zip file, you supply data files only for the business objects that you're loading.

This figure shows an example .zip file structure. The .zip file contains no folders for data (.dat) files.



You define the name of the .zip file, which must be in alphanumeric characters (a-z, A-Z, and 0-9). The .zip file can be encrypted as a whole, but HCM Data Loader doesn't support the encryption of individual data files.

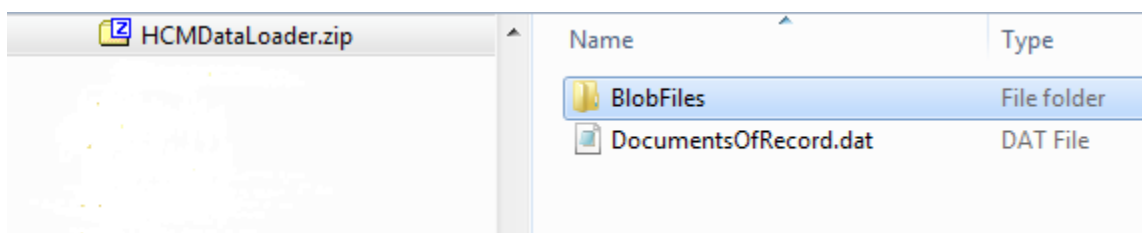
## BlobFiles and ClobFiles Folders in the .zip File

The only two folders that can appear in the .zip file are:

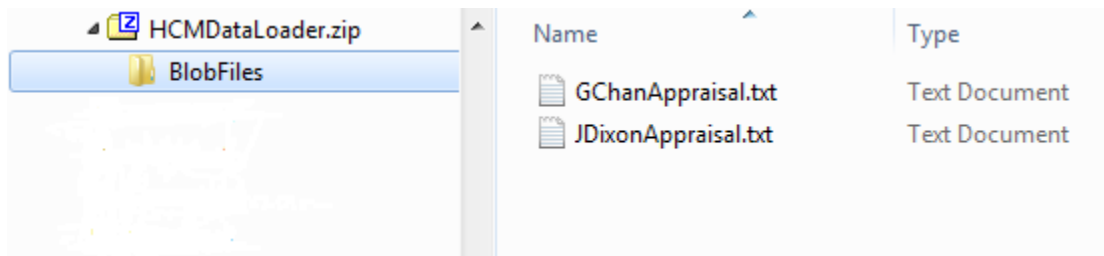
- BlobFiles, which is for binary large objects
- ClobFiles, which is for character large objects

You use these folders for files you want to load as attachments or into large objects. The file names of these attachments or images must be in alphanumeric characters (a-z, A-Z, and 0-9). The data type of the attribute that's used to load your attachment or large-object data determines which folder to use. For example, the **File** attribute in the Document Record object is used for loading attachment files and has a data type of BLOB. Therefore, you place files to be loaded as attachments to document records in the BlobFiles folder.

This figure shows a .zip file structure when attachments are included.



This figure shows the contents of the BlobFiles folder.



## Data Files

Each HCM Data Loader data file is specific to a business object. You provide all components that you want to load for a business-object hierarchy in the same data file. For example, you provide Job, Job Valid Grade, Job Evaluation, Job Extra Information, and Job Legislative Extra Information components in Job.dat. This approach lets you to see all data for a business object and supports the validation of parent references on all child components. It also simplifies the construction of the .zip file, as separate directories aren't required and fewer data files have to be generated.

**Tip:** You can generate data-file templates for all supported objects on the View Business Objects page. Each template defines the complete structure of the selected object.

## Third Party Data Files

You can use third party .zip files to load data with HCM Data Loader, such as pay slips that were generated by a payroll application. The data file must conform to the requirements for the business object you want to load, and the .zip file must have been compressed with `java.util.zip`. If not, then you must extract the files and compress again using `java.util.zip` for the load to work properly.

Here's a tutorial on creating and maintaining data with HCM Data Loader [Create and Maintain Data with HDL](#)

To create and upload a HDL worker.dat file for new hires, see: [Create and Load New Hires with HCM Data Loader \(HDL\)](#)

## What are the various file line instructions used in an HDL file?

This table identifies the instructions that can appear in an HCM Data Loader data file.

Instruction	Line Type	Description
METADATA	Definition	Identifies the business-object component and the attributes of that component for which you're including values in the data file
MERGE	Data	Provides data to be merged in Oracle HCM Cloud

Instruction	Line Type	Description
DELETE	Data	Identifies a business-object component to be purged from Oracle HCM Cloud
SET	Control	Overrides aspects of the default processing for a file
COMMENT	Comment	Adds a comment to the data file

## What are file discriminators?

METADATA, MERGE, and DELETE lines must include a file discriminator value, which identifies the business-object component to which the instruction applies. For example, the available file discriminators for the Job business object are shown in this table.

Business-Object Component	File Discriminator
Job	Job
Job Valid Grade	JobValidGrade
Job Evaluation	JobEvaluation
Job Extra Information	JobExtraInfo
Job Legislative Extra Information	JobLegislative

The file discriminator appears immediately after the instruction in a file line. HCM Data Loader templates for a business object include the file discriminator for each of the object's components. This example shows instructions, the file discriminator, attribute names, and attribute values for the Job business object.

```
METADATA | Job | EffectiveStartDate | EffectiveEndDate | SetCode | JobCode | Name
MERGE | Job | 1950/01/01 | 4712/12/31 | COMMON | CFO | Chief Financial Officer
MERGE | Job | 1950/01/01 | 2013/01/01 | COMMON | PM | Manager Projects
MERGE | Job | 2013/01/02 | 4712/12/31 | COMMON | PM | Manager Projects
MERGE | Job | 1950/01/01 | 4712/12/31 | COMMON | VP_OF_HR | VP Human Resources
MERGE | Job | 1950/01/01 | 4712/12/31 | COMMON | PROG_MGR | Program Manager
```

In this example:

- The instruction, METADATA or MERGE, appears at the start of the line.
- The discriminator on all lines is Job.
- Attribute names appear in the METADATA line after the Job discriminator and separated by the vertical bar character (|).

- Attribute values appear in MERGE lines after the Job discriminator and separated by the vertical bar character (|).

## How to configure METADATA instruction?

The METADATA instruction in an HCM Data Loader data file identifies a business-object component and its attributes. The attributes are those for which you're including values in the data file.

### Instruction Format

The METADATA line has this format:

```
METADATA | <DISCRIMINATOR> | <ATTRIBUTE 1> | <ATTRIBUTE 2> | <ATTRIBUTE n>
```

For example:

```
METADATA | Job | SetCode | JobCode | JobFamilyName | JobName | EffectiveStartDate | EffectiveEndDate
```

### Restrictions

These restrictions apply to the METADATA instruction:

- The METADATA line must appear before the MERGE or DELETE data line to which it relates.
- You can include multiple METADATA lines in a single data file. However, each line must be for a different business-object component. For example, in a Job.dat file, you can include METADATA lines for the Job and Job Valid Grade components. However, you can't include two Job METADATA lines.

### Validation

Each METADATA line must:

- Include a valid discriminator for the object that's identified in the data file name. For example, in a Job.dat file, the METADATA line must include one of these discriminators: Job, JobGrade, JobEvaluation, JobExtrainfo, and JobLegislative.
- Be unique in the file. That is, you can't provide multiple METADATA lines for the same discriminator in a single data file.  
  
For example, having two METADATA lines for the JobGrade discriminator in the same data file is invalid.
- Include only valid attributes for the discriminator. Attribute names are case-sensitive, but they can appear in any order.
- Include the attributes for at least one of the supported key types. For example, if you're using source keys to identify a record uniquely, then you must include SourceSystemOwner and SourceSystemId attributes in the METADATA line.

HCM Data Loader business-object templates include METADATA lines with all available attributes for each component of a business object.

## How to configure MERGE instruction?

The MERGE instruction identifies a data line. You use the MERGE instruction whether creating or updating objects, and HCM Data Loader identifies the correct action.

You can supply either partial or complete objects using the MERGE instruction. The data supplied on the MERGE line is merged into existing Oracle HCM Cloud data. The order in which you specify MERGE lines isn't necessarily the order in which HCM Data Loader processes them.

### Instruction Format

The MERGE line has this format:

```
MERGE |<DISCRIMINATOR>|<ATTRIBUTE 1>|<ATTRIBUTE 2>|<ATTRIBUTE n>
```

For example:

```
MERGE |Job|SE|Software Engineer|Common|2015/01/01|4712/12/31
```

### Restrictions

These restrictions apply to the MERGE line:

- You must not supply multiple MERGE lines for the same component in a single data file, unless the object is date-effective. For example, you can't supply two MERGE lines, one to create a Person Ethnicity component and one to update it, in the same data file. As the component isn't date-effective, these instructions would correct the data rather than update it. In such cases, you can supply only current data. Alternatively, you can create the record in one data file and update it in a different data file.

**Note:** HCM Data Loader doesn't process individual file lines, but groups related lines. This grouping works for date-effective records because the file lines are processed in order of effective start date.

- You can include MERGE and DELETE instructions in the same file, but not for the same logical record. For example, you can't load multiple date-effective records for the same component, such as a single Job, using a mix of MERGE and DELETE instructions. However, you can provide both MERGE and DELETE instructions for components of a single logical object. For example, you can provide:
  - MERGE instructions for Worker and Person Name
  - DELETE instructions for the Person Email and Person Phone components of the same Worker object

### Validation

Each MERGE line must:

- Be preceded by the METADATA line for the same discriminator.
- Contain the same attributes as the relevant METADATA line and in the same order.
- Not include values for attributes that aren't in the METADATA line.

- Contain a unique identifier for itself, using any of the supported key types. In addition, each data line for a child component must include a unique reference to its parent. For example, GradeRateValue must include a unique reference to its associated GradeRate.

## How to configure DELETE instruction?

The DELETE instruction identifies a business-object component to purge from Oracle HCM Cloud. The component doesn't have to have been loaded using HCM Data Loader.

### Instruction Format

The DELETE instruction has this format:

```
DELETE | <DISCRIMINATOR> | <ATTRIBUTE 1> | <ATTRIBUTE 2> | ATTRIBUTE n>
```

For example:

```
METADATA | JobFamily | EffectiveStartDate | EffectiveEndDate | JobFamilyName  

DELETE | JobFamily | 2012/10/01 | 4712/12/31 | Sales01
```

### Restrictions

These restrictions apply to the DELETE instruction:

- You can delete a business object or a component of a business object. However, you can't delete individual date-effective records.
- You can't provide MERGE and DELETE instructions for the same record in a single data file. The order in which HCM Data Loader processes instructions is unpredictable.
- You can't recover deleted records. Therefore, try to correct data rather than delete and recreate it.
- Not all business-object components can be deleted. For information about deletion support, see the business-object details available from the View Business Objects page in the Data Exchange work area.

## How to configure SET instruction?

The SET instruction is optional in a .dat file and has no discriminator. You use SET instructions to alter default processing for the file in which they appear.

You can include any number of SET instructions in a .dat file, but they must appear before the first METADATA line.

This table lists the SET instructions and their default values.

Instruction	Default Value
SET PURGE_FUTURE_CHANGES Y N	Y
SET DISABLE_POST_PROCESS_TASKS <process>	Not applicable

Instruction	Default Value
SET INVOKE_POST_PROCESS Y N	Y
SET FILE_DELIMITER <delimiter>	Vertical bar ( )
SET FILE_ESCAPE <escape character>	Backslash (\)
SET FILE_NEWLINE <newline character>	n
SET ENABLE_AUDIT_DATA Y N	N
SET PURGE_AUDIT_DATA Y N	N
SET CALCULATE_FTE Y N	N
SET CREATE_DEFAULT_WORKING_HOUR_PATTERN Y N	N
SET ENABLE_INCREMENTAL_LOAD_EVENTS Y N	N
<p><b>Note:</b>                      Enabling events for your file may increase the time it takes to load.</p>	
SET EXTEND_DATA_SET_RETENTION Y N	N
<p><b>Note:</b>                      Extending the retention period of a data set will lengthen the time it takes to load, as after processing the staging table data is moved from the primary to the secondary staging tables.</p>	
SET EVALUATE_LIFE_EVENTS Y N	N
SET RETAIN_SUCCESSION_PLAN_HISTORY Y N	Y
SET ORA_ENABLE_OS_BULK_INGEST Y N	Y

Instruction	Default Value
SET ORACLE_SEARCH_UPDATE_MODE ORA_BULK ORA_INITIAL	ORA_BULK

**Tip:** The default values shown here are predefined for the enterprise. You can override these values for the enterprise using the **Configure HCM Data Loader** task in the Setup and Maintenance work area.

## SET Instruction Descriptions

This section provides an alphabetic list of all SET instructions with their descriptions.

### SET CALCULATE\_FTE

To request automatic calculation of full-time equivalent (FTE) values for all worker assignments in a Worker.dat file, include the SET CALCULATE\_FTE instruction. For example:

```
SET CALCULATE_FTE Y
```

### SET CREATE\_DEFAULT\_WORKING\_HOUR\_PATTERN

To request automatic creation of a default working-hour pattern for all worker assignments in a Worker.dat file, include the SET CREATE\_DEFAULT\_WORKING\_HOUR\_PATTERN instruction. For example:

```
SET CREATE_DEFAULT_WORKING_HOUR_PATTERN Y
```

### SET DISABLE\_POST\_PROCESS\_TASKS

By default, these processes run automatically after you load workers:

- Refresh Manager Hierarchy
- Update Person Search Keywords

To prevent **Refresh Manager Hierarchy** from running automatically, you can include this SET instruction in the Worker.dat file:

```
SET DISABLE_POST_PROCESS_TASKS RefreshManagerHierarchy
```

To prevent **Update Person Search Keywords** from running automatically, you can include this SET instruction in the Worker.dat file:

```
SET DISABLE_POST_PROCESS_TASKS UpdatePersonSearchKeywords
```

By default, the **FlattenPositionHierarchy** process runs automatically after you load HCM position hierarchies. To prevent **FlattenPositionHierarchy** from running automatically, include this SET instruction in the PositionHierarchy.dat file:

```
SET DISABLE_POST_PROCESS_TASKS FlattenPositionHierarchy
```

**Tip:** To prevent postload processes from running automatically, you can use the SET INVOKE\_POST\_PROCESS instruction instead.

### SET ENABLE\_AUDIT\_DATA

Audit of objects loaded using HCM Data Loader is enabled or disabled for the enterprise using the **Enable Audit Data** configuration parameter. You can override the enterprise setting for individual .dat files by including this SET instruction. For example, if audit is enabled for the enterprise, then you can disable it for an individual .dat file.

### SET FILE\_DELIMITER

Use this instruction to change the default file delimiter, which is the vertical bar (|). For example, to change it to a comma, include this instruction:

```
SET FILE_DELIMITER ,
```

The new value can be up to 10 characters.

### SET FILE\_ESCAPE

Use this instruction to change the default file escape character, which is the backslash (\). For example, to change it to the equal sign (=), include this instruction:

```
SET FILE_ESCAPE =
```

The new value can be up to 10 characters.

### SET EXTEND\_DATA\_SET\_RETENTION

By default, all data sets are automatically and permanently deleted after a maximum of 30 days since they were last updated. If you are loading data which is to be validated, or may need to be rolled back, you can use this instruction to extend the retention period of the data set.

Supply this instruction only in business object files that support validate or roll back. If you are supplying multiple business object files that support validate or roll back, you need only supply this instruction once.

```
SET EXTEND_DATA_SET_RETENTION Y
```

**Note:** The time it takes to complete the processing of your file will be longer, as after loading the valid data into the application tables, the staging table data is moved from the primary to the secondary staging tables, so it doesn't negatively impact the processing time of subsequent files.

### SET FILE\_NEWLINE

Use this instruction to change the default file newline character, which is the character `\n`. For example, to change it to the string `newline`, include this instruction:

```
SET FILE_NEW_LINE newline
```

The new value can be up to 10 characters.

### SET INVOKE\_POST\_PROCESS

By default, these processes run automatically after you load workers:

- Refresh Manager Hierarchy
- Update Person Search Keywords

By default, the **FlattenPositionHierarchy** process runs automatically after you load HCM position hierarchies.

To prevent such processes from running automatically, you can include this SET instruction in the relevant .dat file:

```
SET INVOKE_POST_PROCESS N
```

## SET PURGE\_AUDIT\_DATA

Use this instruction to purge the audit data for the objects in a .dat file. This feature is intended primarily to remove the audit records for personally identifiable information in worker records.

You can't include both of the following instructions in a single .dat file:

- `SET ENABLE_AUDIT_DATA Y`
- `SET PURGE_AUDIT_DATA Y`

If auditing of loaded objects is enabled for the enterprise, then you must disable it in any .dat file where you're purging audit data. That is, you include the following instructions:

- `SET ENABLE_AUDIT_DATA N`
- `SET PURGE_AUDIT_DATA Y`

## SET PURGE\_FUTURE\_CHANGES

When you create business objects containing date-effective components, you include this instruction at the start of the file:

```
SET PURGE_FUTURE_CHANGES Y
```

This instruction sets the date-effective maintenance mode to Replace. In this mode, which is the default mode, the contents of the .dat file replace any existing data.

When you update business objects containing date-effective components, you include this instruction at the start of the file:

```
SET PURGE_FUTURE_CHANGES N
```

This instruction sets the date-effective maintenance mode to Retain. In this mode, any existing future-dated changes are retained.

## SET ENABLE\_INCREMENTAL\_LOAD\_EVENTS

To enable the firing of atom feeds and events that are triggered by the changes supplied in a file, include the `SET ENABLE_INCREMENTAL_LOAD_EVENTS` instruction. For example

```
SET ENABLE_INCREMENTAL_LOAD_EVENTS Y
```

**Note:** Events are only triggered if they are enabled in your environment. Enabling events for your file may increase the time it takes to load.

## SET RETAIN\_SUCCESSION\_PLAN\_HISTORY

Determines whether historical data is retained for changes to succession plans. To prevent the retention of history, include the `SET RETAIN_SUCCESSION_PLAN_HISTORY` instruction in your SuccessionPlan.dat files. For example:

```
SET RETAIN_SUCCESSION_PLAN_HISTORY N
```

## SET ORA\_ENABLE\_OS\_BULK\_INGEST

To automatically submit the **Update Oracle Search with HCM Data Loader Changes** process to maintain the Oracle Search indexes with the changes made.

```
SET ORA_ENABLE_OS_BULK_INGEST Y
```

## SET ORA\_UPDATE\_ORACLE\_SEARCH

To enable bulk-ingestion include the following instruction at the top of your file:

```
SET ORA_UPDATE_ORACLE_SEARCH ORA_BULK
```

It enables the capture of Fusion surrogate IDs for Oracle Search bulk-ingestion and the automated submission of the **Update Oracle Search with HCM Data Loader Changes** process.

## How do I generate a single template file?

HCM Data Loader provides a template file for each supported business-object hierarchy. The template contains:

- A COMMENT line, which identifies the business object, its version, and the file-creation date.
- A METADATA line for each component of the business-object hierarchy that you can load for the business object. The METADATA line includes every attribute of the component, including environment-specific configured flexfield attributes.

You can generate template files and use them as the basis of your own data files.

Follow these steps to generate single template files:

1. Open the **Data Exchange** work area and select **View Business Objects**.

By default, the View Business Objects page lists all business objects that HCM Data Loader supports. You can sort the list by product area, business object, or load order, for example. You can also retrieve a specific object by entering its name in the **Business Object** query-by-example field and pressing **Enter**.

2. If a template already exists for the object, then you can click the icon in the **File** column to download it. The **Last Refreshed** column shows when the object definition was last updated and the template was last generated. Alternatively, you can update the object definition before you download the template, as follows:
  - a. Select the object row in the business objects table.
  - b. Click **Refresh Object**.
  - c. In the Schedule Request dialog box, click **Submit**. Click **OK** to close the confirmation message.

When you click **Submit**, you start the process that updates the object definition and generates the latest template file.

**Tip:** To monitor progress, click **View Process Results** to open the View Process Results page.

3. On the **View Business Objects** page, click the **Refresh** icon.

Check the value in the **Last Refreshed** column to confirm that the template has been updated.

4. Click the icon in the **File** column to download the template.

**Tip:** You can download the complete object definition for offline use. To download the object definition, click the icon in the **Overview** column. The spreadsheet that you download is documentation for the object, not a template file.

## How do I generate template files for all business objects?

HCM Data Loader provides a template file for each supported business-object hierarchy. The template contains:

- A COMMENT line, which identifies the business object, its version, and the file-creation date.
- A METADATA line for each component of the business-object hierarchy that you can load for the business object. The METADATA line includes every attribute of the component, including environment-specific configured flexfield attributes.

You can generate template files and use them as the basis of your own data files.

Follow these steps to generate template files for all business objects in a product area or all business objects:

1. Click the **Refresh All Objects** button on the View Business Objects page.
2. In the Schedule Request dialog box, you can either select a product area or leave **Product Area** set to **All**.
3. Click **Submit**.
4. When the generation process completes, you can download templates individually. Click the icon in the **File** column for an object on the View Business Objects page. Alternatively, follow these steps to download a single .zip file of the generated templates:
  - a. Click View Process Results.
  - b. On the View Process Results page, click the icon in the **File** column for your process.

## How do I use template files?

Template files identify the names of all available attributes of a business object, including those of environment-specific attributes, such as flexfield segments. Don't simply copy the METADATA lines from the template file and use them in your own data files. Instead, include in your data files:

- METADATA lines for the components that you want to load
- Attribute names for the values you want to supply

You can improve the speed and efficiency of the import and load processes by including only the attributes for which you're supplying data. Remove METADATA lines and attribute names that you don't plan to use.

HCM Data Loader validates every attribute name on every METADATA line.

## How do I view the structure, supported key systems, data formats, and required attributes for business objects?

The **View Business Objects** table lists all business object available for bulk-loading.

From here you can -

- Download business object template files.
- Download the complete definition of an object for offline use.
- Identify which objects support being rolled back.
- Review the object hierarchy and supported attributes for an object.

You're recommended to refresh all objects after an update. The refresh ensures that any changes from the update appear on the View Business Objects page and in object templates. If no details exist for a selected object when you view it, then you're prompted to refresh the object.

You can refresh business objects using the Refresh Object icon and the Refresh All Objects button. When you refresh an object:

- Attribute and flexfield information for the selected business objects is updated.
- Up-to-date templates of **METADATA** lines are generated for the selected business objects.
- The Overview spreadsheet can be downloaded for the selected business object.
  - a. Click the icon in the Overview column to download the complete definition of an object for offline use. The default file name of the downloaded spreadsheet is GenericBusObjDetails.xlsx, but you can change the name when you save the file.

**Note:** Whenever you open a downloaded spreadsheet, you're invited to sign in again. If you sign in, then the spreadsheet is updated automatically with latest information as you select each tab. These updates overwrite any changes that you may have made to the spreadsheet. If you prefer to keep your changes, then don't sign in.

When you click the name of a business object on the View Business Objects page, you access information about the object. This information includes:

- The component hierarchy of the object.
- Component details for the currently selected component. These include:
  - Its parent component, file discriminator, .dat file, and translation object
  - Whether it supports integration keys
  - Its supported actions, for example, whether you can delete the component
  - Whether it's date effective and, if so, whether you can change the component's first start date and last end date
- Attribute details for each attribute of the currently selected component. These include:
  - Its name, type, and length
  - A translated label and description
  - Whether it's required always or for new objects only
  - Whether it's a key value or can be used as a key value and, if so, the key type
  - For attributes that are keys of type surrogate ID, parent surrogate ID, or foreign-object reference, whether integration keys are also supported
  - The names of any user-key attributes that you can use in place of surrogate IDs
  - Names of lookup types, for attributes that are lookup-validated
- Flexfield attribute details for each of the configured and deployed flexfield segments. These include:

- The attribute name, flexfield segment label and description
- The data type, length, and required status
- The value set used to validate the flexfield segment
- **Note:**
  - Two attributes are supplied for flexfield segments, which are validated by a value set. The attribute that includes `_Display` in the name accepts the meaning, and the other attribute accepts the ID.
  - Some components restrict the flexfield contexts you can bulk-load using HCM Data Loader. Hence, you may not see a flexfield context you created.
- On the **View Business Objects** page you will now see a red alert icon against object hierarchies that are being withdrawn and a yellow warning icon against object hierarchies that include components that are being withdrawn. You can also see icons against components in the business object hierarchy that are always or conditionally required when creating new records.

If you can load flexfields for the component, then the Flexfield Attributes tab appears. For descriptive flexfields and extensible flexfields, you can see the attributes for a selected flexfield and context. For extensible flexfields, category also appears. When a flexfield context supports multirow entries, one or more flexfield attributes are identified as user keys to reference a single record uniquely. These attributes are identified in the **User Key** column on the Flexfield Attributes tab. This column doesn't appear if the flexfield doesn't support multi-row entries.

You can refresh business objects using the **Refresh Object** icon and the **Refresh All Objects** button. When you refresh an object:

- Attribute and flexfield information for the selected business objects is updated.
- Up-to-date templates of METADATA lines are generated for the selected business objects.

## How can I set up the configuration parameters for HCM Data Loader and HCM Spreadsheet Data Loader import and load processes?

The HCM Data Loader configuration parameters control the HCM Data Loader and HCM Spreadsheet Data Loader import and load processes. The default settings of the parameters suit most requirements.

You can override these settings to suit the configuration of your environment and the amount of data being loaded. Most enterprise settings can be overridden for individual load processes.

In the **Setup and Maintenance** work area, navigate to **Configure HCM Data Loader** task under the **HCM Data Loader** functional area to review or override the predefined settings for the enterprise:

You can filter the parameters using the following category values:

- Availability
- Diagnostics
- File Definition
- Performance

- Scheduling Default

Parameters that are specific to a business object appear in a separate section.

**Note:** HCM Spreadsheet Data Loader has its own versions of some configuration parameters, which you manage on the **HCM Spreadsheet Data Loader** tab. For these parameters, you can set different default values for HCM Data Loader and HCM Spreadsheet Data Loader. Users can override the HCM Spreadsheet Data Loader default values in spreadsheet templates.

**Note:** A **More Information** icon is introduced on the names of the parameters which can be overridden by a SET instruction. When you click the orange icon, the **More Information** pop up appears, and you will see the details of the SET instruction that overrides the data loading behavior of your file.

## What are the various HCM Data Loader configuration parameters?

### Availability Parameters

#### Enable Automatic Extended Data Set Retention

Determines whether data sets for business objects that support validate or rollback are automatically retained beyond the standard retention period. You can override the default behavior using a SET instruction. You control how long extended retention data sets are kept for using the Days to Preserve Extended Retention Data Sets parameter.

Default Value: No

#### Enable Audit Data

Specifies whether audit data is captured for business objects loaded using HCM Data Loader. If you set this parameter to **Yes**, then audit must also be enabled for Oracle Fusion Applications.

You can set this parameter separately for HCM Data Loader and HCM Spreadsheet Data Loader.

Default Value: No

#### Enable Bulk-Ingestion of Oracle Search Indexes

Enable changes uploaded by HDL and HSDL to be bulk-ingested into Oracle Search indexes. When disabled, incremental ingestion is used.

Default Value: Yes

#### Enable Configuration of Role-Based Business Object Access

Determines whether access to HCM Data Loader business objects is configurable by role. When enabled you must configure the business objects a custom role can load data with.

Default Value: No

#### Enable File Generation for Failed File Lines

Determines whether you can generate business object files for the files lines that failed to load.

Default Value: No

### Enable HCM Data Loader Post Process Configuration

Determines whether configuration of the processes to be automatically submitted after a data load is enabled.

Default Value: Yes

### Enable Load Events for Incremental Updates

Enable events to be triggered for the data supplied. Enabling load events will significantly increase the time it takes for your data to load. You can override this parameter with a set instruction followed by the override value. Place SET instructions before the first METADATA line in your business object file.

Default Value: No

### Enable Load Group Size Calculation

Specifies whether the load-group size is to be calculated automatically. Set to **Yes** to enable the optimal load-group size to be calculated based on the number of objects to process and the available concurrent threads. When **Enable Load Group Size Calculation** is **Yes**, the **Load Group Size** parameter doesn't appear on the Schedule Request page. Also, the default value of the **Load Group Size** parameter is ignored.

Default Value: Yes

### Enable UI Message Translation

Specifies whether messages on the Import and Load Data page are to appear in the language of the signed-in user. Set to **Yes** to enable translated messages. If this parameter is set to **No**, then messages appear in the language of the user who submitted the import and load request.

Default Value: No

### Generate Business Object Mapping Files

Specifies one of three options for generating Business Object Mapping Files. The mapping file describes the HCM Data Loader shape, and you can import this into Oracle Integration Cloud to define transformation and mappings rules.

The default value of **No** means you haven't enabled the option to generate mapping files.

The **Custom only** value lets you generate mapping files for business object dat files you upload to HCM Data Loader. Only the attributes included in the uploaded file will be included in the mapping file. Under the Parameters section, select the Generate Objects Mapping File feature. Refresh the Business Objects task bar until the Download Mapping File link activates and then you can retrieve the map.

A parameter value of **Yes** lets you request a mapping file from the Import and Load Data interface, just like Custom only. In addition, on the View Business Objects page, you can Refresh an individual object and it will automatically generate a mapping file for you. Refresh the page until Last Refreshed indicates today's date. Download the mapping file using the link under Integration Mapping File.

Default Value: No

### Purge Person Enabled Key

Key value supplied by Oracle Support to enable the **Purge Person Data in Test Environments** process.

You can't enable and run this process in production environments.

Default Value: None

### **Enable Load Events for Incremental Updates**

Specifies whether incremental load events are enabled when bulk-loading data.

Default Value: No

**Note:** You can't override the default value for the enterprise. You can include a SET instruction in individual .dat files to enable incremental load events for the objects in the files.

### **Enable Mandatory Descriptive Flexfield Segment Validation**

Determines whether an error is raised when loading descriptive flexfield data, a mandatory segment is without a value, and no default is configured.

Default Value: No

### **Restrict Access to Security Related Business Objects**

Determines whether users with a role granted with the Load HCM Security Data function security privilege can bulk-load data using security related objects. When set to Yes, only users with a role that is granted the Load HCM Security Data function security privilege will be able to bulk-load data using security related objects, such as User and Delegated Role. When set to No, users that aren't granted the Load HCM Security Data function security privilege will encounter warnings when loading security-related data, but the uploads will continue.

**Note:** Once you enable this feature you can't disable it later. This functionality will be enabled by default in a future release.

For more information, refer [How You Enable Access to HCM Data Loader](#).

## **Diagnostics Parameters**

### **Complex Error Stack Trace Occurrences by Thread**

Maximum number of complex error message occurrences on a processing thread for which stack trace is recorded.

Default Value: 5

### **Data Error Stack Trace Occurrences by Thread**

Maximum number of data error message occurrences on a processing thread for which stack trace is recorded.

Default Value: 2

## **File Definition Parameters**

### **Data Set Name**

For data loads initiated by web service, specifies whether the Oracle WebCenter Content document title or the .zip file name is used as the data set name.

Default Value: File name

### **File Character Set**

Character set used for business object and attachment files.

Default Value: UTF-8

### **File Delimiter**

A value of up to 10 characters used to separate attribute names on METADATA lines and attribute values on data lines.

Default Value: Vertical bar (|)

### **File Encryption**

Default file encryption.

Default Value: None

### **File Escape Indicator**

A value of up to 10 characters used to indicate that the next character is to be ignored on data lines.

Default Value: Backslash (\)

### **File New Line Indicator**

A value of up to 10 characters used to indicate a new line in an attribute value. This value must be preceded by the file escape indicator.

Default Value: n

### **Flexfields Enabled**

Controls whether flexfield data can be loaded.

Default Value: Yes

### **Purge Audit Data**

Specifies whether audit data is to be purged automatically for objects successfully updated or deleted.

Default Value: No

**Note:** You can't override the default value for the enterprise. You can include a SET instruction in individual .dat files to purge audit data for the objects in the files.

## **Performance Parameters**

### **Days to Preserve Standard Data Sets Before Archiving**

Number of days since a standard data set was last updated before it can be automatically archived. Data sets are only archived if the number of stage table lines exceed the configured limit.

Default Value: 5

### **Days to Preserve Standard Data Sets Before Deleting**

The number of days since a standard data set was last updated before it's automatically and permanently deleted from the stage tables.

Default Value: 30

### **Days to Preserve Extended Retention Data Sets**

Number of days since an extended retention data set was last updated before it's automatically deleted from the stage tables. Only data sets that contain data which can be validated or rolled back can be retained for extended periods.

Default Value: 60

### **Load Group Size**

Number of business objects processed as a single unit of work by a single thread.

Default Value: 100

### **Maximum Concurrent Threads for Import**

Maximum number of threads to run concurrently when importing data to the stage tables.

You can set this parameter separately for HCM Data Loader and HCM Spreadsheet Data Loader.

Default Value: 8

### **Maximum Concurrent Threads for Load**

Maximum number of threads to run concurrently when loading data from the stage tables to the application tables.

This parameter is used in conjunction with the **Load Group Size** parameter. Only the threads required to load your objects based on the **Load Group Size** parameter are created.

Default Value: 8

### **Maximum Data Lines Before Archiving**

The number of data lines that can be held in the stage tables before data sets are automatically backed up.

Default Value: 10000000

### **Transfer Group Size**

Number of file lines processed as a single unit of work by a concurrent thread.

Default Value: 100000

## **Scheduling Default Parameters**

### **Date-Effective Update Mode**

Specifies whether future-dated records should be purged when date-effective objects are loaded.

You can set this parameter separately for HCM Data Loader and HCM Spreadsheet Data Loader.

Default Value: Replace

### **Delete Source File**

Specifies whether to delete source files from the Oracle WebCenter Content server once they have been processed.

Default Value: Yes

### **Disable Post Process Tasks**

This parameter is read-only. You can disable individual postload processes by including a SET instruction in the relevant Worker.dat or Position.dat file.

Default Value: None

### **File Action**

Default file-processing action.

**Import and load** means that valid objects are loaded automatically once import completes. **Import only** means that your data is imported to the stage tables, but load isn't initiated.

Default Value: Import and load

### **Initiate Business Object Post Processing**

Specifies whether processes registered to run automatically after an object is loaded should run.

You can set this parameter separately for HCM Data Loader and HCM Spreadsheet Data Loader.

Default Value: Yes

### **Maximum Percentage of Import Errors**

Percentage of file lines in error that can occur in a business object before the import process stops for the object.

Default Value: 100

### **Maximum Percentage of Load Errors**

Percentage of business-object instances in error that can occur for a business object before the load process stops.

You can set this parameter separately for HCM Data Loader and HCM Spreadsheet Data Loader.

Default Value: 100

## **Business Object Parameters: Worker**

**You can set these parameters separately for HCM Data Loader and HCM Spreadsheet Data Loader.**

### **Calculate Worker Full-Time Equivalent**

Specifies whether the full-time equivalent (FTE) value should be calculated automatically for all worker assignments in a Worker.dat file.

Default Value: No

### **Create Worker Default Work Hour Pattern**

Specifies whether a default working-hour pattern should be created automatically for all worker assignments in a Worker.dat file.

Default Value: No

### **Enable Worker Address Validation**

Specifies whether a default address validation should be applied for all worker assignments in a Worker.dat file.

Default Value: Yes

## Business Object Parameters: Location

### Enable Location Address Validation

Specifies whether a default address validation should be applied for all locations in a Location.dat file.

Default Value: Yes

## Business Object Parameters: Participant Enrollment

### Evaluate Life Events

Determines whether life events should be evaluated when loading Participant Enrollment data.

Default Value: No

**Note:** You can override the value of this parameter using the **Configure HCM Data Loader** task in the Setup and Maintenance work area.

## Business Object Parameters: Succession Plan

### Retain Succession Plan History

Determines whether historical data is retained for changes to succession plans.

You can set this parameter separately for HCM Data Loader, HCM Spreadsheet Data Loader and HCM Spreadsheet Data Loader templates.

Default Value: Yes

**Note:** You can override the value of this parameter using the **Configure HCM Data Loader** task in the Setup and Maintenance work area.

## How do I audit the changes done to HCM Data Loader parameters?

Monitor changes to the parameters that control the behavior and availability of bulk-loading functionality by enabling an audit of the parameters available on Configure HCM Data Loader.

### To enable audit for Configure HCM Data Loader:

1. Perform the **Manage Audit Policies** task in the **Setup and Maintenance** work area.
2. Configure the audit parameters for **Oracle Fusion Applications**.
3. Select **HCM Common Architecture** as the **Product**.
4. Check the checkbox for **Configure HCM Data Loader Parameters** and select the node in the hierarchy.
5. Add attribute **Parameter Value**.
6. Save your changes.

Changes made using the **Configure HCM Data Loader** task will now be audited.

**To review the audit report for Configure HCM Data Loader:**

1. Navigate to **Tools > Audit Reports**.
2. Select the **Product** as **HCM Common Architecture** and the **Business Object Type** as **Configure HCM Data Loader Parameters**.
3. Select a date and click **Search**.
4. On the table toolbar, check the **Show Attribute Details** and **Show Additional Object Identifier Columns** checkboxes.
5. Select **All Attributes** within the **Show Attribute Details** choice list.

## How do I define referenced objects?

Business objects that you're loading may reference a few business objects that you can't load using HCM Data Loader because they're not integration enabled. You define or review these objects in the target Oracle HCM Cloud environment before you load data that references them.

You may have performed this step during implementation of Oracle HCM Cloud.

This table identifies the main objects of this type. It lists the tasks that you use to manage them and the functional area to which each task belongs in the Setup and Maintenance work area.

Business Object	Functional Area	Task
Application Reference Data Set	Enterprise Profile	Manage Reference Data Sets
Assignment Status Type	Workforce Information	Manage Assignment Status
Business Unit	Organization Structures	Manage Business Unit
Content Type	Workforce Profiles	Manage Profile Content Types
Currency Code	Financial Reporting Structures	Manage Currencies
Element Type	Elements and Formulas	Manage Elements
Legal Entity	Legal Structures	Manage Legal Entity
Official Language Code	Application Extensions	Manage Languages
Profile Type	Workforce Profiles	Manage Profile Types
Schedule	Workforce Information	Manage Work Schedules

## Foreign-Object References

When you refer to these objects from objects that you're loading, you use their user keys. (Alternatively, you can use their surrogate IDs, if available.) HCM Data Loader provides business-object documentation for all supported objects. This documentation identifies the user key that you can use for foreign-object references. For example, the Position object includes a reference to the Business Unit object, which isn't integration enabled. The position documentation identifies the business-unit name as its user key. Therefore, when loading a Position object, you can refer to the associated business unit using the business-unit name. To see the supplied documentation for a business object, click the object name on the View Business Objects page.

## How do I review list of values business object attributes?

The permitted values of many object attributes are defined in lists of values. Some lists of values are predefined and can't be updated in any way. Others contain some values, but you can also add your own.

In some cases, you can edit or remove predefined values. In Oracle HCM Cloud, lists of values are defined as lookup types. You're recommended to review the predefined lookup types and make any updates before you attempt to load data that uses them. You may have completed this process during implementation of Oracle HCM Cloud.

## Reviewing Lookup Types

To manage lookup types, perform relevant tasks in the Setup and Maintenance work area. For example, the Workforce Information functional area contains tasks for managing many lookup types, including person lookups, termination lookups, document lookups, and checklist lookups. To manage person lookup types, perform the **Manage Person Lookups** task. On the Manage Person Lookups page, select a lookup type to edit. The list includes:

- Address types
- Contact relationships
- Ethnicity
- Highest education level
- Honors
- Marital status
- Military rank
- Phone types
- Religion

Ensure that the defined lookup values support the data that you're planning to load.

## How do I define the source-system owner values?

If you plan to use source keys, then you must add your source-system owner values to the HRC\_SOURCE\_SYSTEM\_OWNER lookup type before you load data.

You must have the Application Implementation Consultant or Human Capital Management Application Administrator job role or privileges to perform this task.

Use the **Manage Common Lookups** task in the Setup and Maintenance work area. Follow these steps:

1. On the Manage Common Lookups page, search for the lookup type **HRC\_SOURCE\_SYSTEM\_OWNER**.
2. In the search results, select the lookup type to display its lookup codes.
3. In the Lookup Codes section of the page, click the **New** icon to add a lookup-code row.
4. Complete the fields in the row. Ensure that the new code is enabled and the dates are valid for the data that you're loading.
5. Repeat steps 3 and 4 for additional source-system owner values.
6. Click **Save and Close**.

**Note:** The source-system owner values are saved in uppercase by default.

## How can I test the HCM Data Loader process flow and connections?

In a new environment, you may want to test HCM Data Loader end-to-end processing without creating unwanted data. Once you're sure that HCM Data Loader processing and connections are working as expected, you can load your own data with confidence.

### Testing End-to-End Processing

HCM Data Loader provides:

- A process, **Test HCM Data Loader Process Flow and Connections**
- An object, Example Object

When you submit the **Test HCM Data Loader Process Flow and Connections** process, set the **Action** parameter to **Test HCM Data Loader**. The process:

1. Generates a file of MERGE instructions for the Example Object
2. Adds the ExampleObject.dat file to the **ExampleObject||<date and time>|.zip** file
3. Uploads the .zip file to the **hcm/dataloader/import** directory on the Oracle WebCenter Content server
4. Initiates HCM Data Loader to import and load the file
5. Provides a log file

You can review progress and results on the Import and Load Data page.

### The Example Object

When you run the **Test HCM Data Loader Process Flow and Connections** process, the Example Object data is saved to its own tables. No data is saved to the application tables. You can review the structure and attributes of the Example Object on the View Business Objects page in the Data Exchange work area. The example object:

- Has child and grandchild components and a translation object
- Supports all key types

- Supports Create and Delete actions
- Has attributes of most data types
- Has lookup-validated attributes

You can generate a template file for the Example Object.

**Note:** Although an Example Object Translation object exists, no translation data is generated when you run the **Test HCM Data Loader Process Flow and Connections** process.

## Deleting the Test Data

You're recommended to run the **Test HCM Data Loader Process Flow and Connections** process periodically to delete the test data and associated data sets. When you submit the process, set the **Action** parameter to **Delete test data**. In this case, the process generates a file of DELETE instructions, adds it to the **DeleteExampleObject||<date and time>|.zip** file, and uploads it.

## How do I enable and manage audits for bulk-loaded objects?

You can audit changes made to objects when you bulk load them using either HCM Data Loader or HCM Spreadsheet Data Loader. When audit is enabled, the creation, update, and deletion of objects are audited.

**Tip:** Enable audit only for incremental updates and not during data migration.

## Enabling Audit for Bulk-Loaded Objects

Auditing is optional. To audit bulk loading of objects, you must:

1. Perform the **Manage Audit Policies** task in the Setup and Maintenance work area to select objects and their attributes for audit.
2. Enable auditing in general on the Manage Audit Policies page.
3. Enable auditing of bulk-loaded data, either for the enterprise or in individual .dat files and spreadsheet templates.

## Enabling Audit for HCM Data Loader

The **Enable Audit Data** parameter on the HCM Data Loader parameters tab of the Configure HCM Data Loader page controls audit for the enterprise. By default, **Enable Audit Data** is set to **No**, but you can override that setting. Alternatively, you can enable audit selectively by including the following SET instruction in relevant .dat files:

```
SET ENABLE_AUDIT_DATA Y
```

## Enabling Audit for HCM Spreadsheet Data Loader

The **Enable Audit Data** parameter on the HCM Spreadsheet Data Loader parameters tab of the Configure HCM Data Loader page controls audit for the enterprise. By default, the **Enable Audit Data** parameter is set to **No**, but you can

override that setting. Alternatively, you can enable audit selectively by setting the **Enable Audit Data** parameter in the relevant spreadsheet template to **Yes**.

## When to Enable Audit for the Enterprise

Setting the **Enable Audit Data** configuration parameters to **Yes** for the enterprise may affect performance. Consider enabling audit in this way only for ongoing integrations.

## Purging Audit Data

You may want to purge audit data, typically when disposing of personally identifiable information in worker records. By default, the purging of audit data is disabled for the enterprise and you can't override this setting at the enterprise level. However, you can include the following SET instruction in individual .dat files:

```
SET PURGE_AUDIT_DATA Y
```

In spreadsheet templates, you can set the **Purge Audit Data** parameter to **Yes**. These instructions purge the audit data for the business objects in the .dat file or spreadsheet.

Restrictions exist on enabling audit and purging data at the same time. You can't include both of the following SET instructions in a single .dat file:

```
SET ENABLE_AUDIT_DATA Y
SET PURGE_AUDIT_DATA Y
```

Similarly, you can't include both of the following parameter values in a spreadsheet template:

Parameter	Value
Enable Audit Data	Yes
Purge Audit Data	Yes

To purge audit data when auditing is enabled for the enterprise, you must disable audit in the same .dat file or spreadsheet template. That is, you must do one of the following:

- Include a `SET ENABLE_AUDIT_DATA N` instruction in the same .dat file.
- Set the **Enable Audit Data** parameter in the spreadsheet template to **No**.

## Correcting Load Errors for Audited Objects

When audit is enabled for an object, you can't correct load errors for it in the HCM Data Loader user interface. For example, your .zip file may contain both Worker and Job objects. If you include the `SET ENABLE_AUDIT_DATA Y` instruction in the Worker.dat file only, then you can't correct load errors from that file in the user interface. However, you can correct errors in the Job objects, as audit isn't enabled in the Job.dat file.

# How do I prepare data for loading successfully into Oracle HCM Cloud?

Follow these rules to ensure successful data upload:

**Reviewing and Cleansing Source Data:** Identify the business objects that you're planning to upload to Oracle HCM Cloud and their source systems. Review this source data, and verify that it's both accurate and current.

Correct any problems before you attempt to extract it. For example:

- Ensure that a manager is identified for every worker and that the information is accurate.
- For jobs and positions, ensure that correct job codes and titles exist in the source systems.
- For worker history, establish the accuracy of any historical data. Understand whether all historical data must be uploaded or just key events, such as hire, promotion, and termination.

**Extracting Data from the Source System:** Define mappings between your source data and the Oracle HCM Cloud business-object model by comparing source and target attributes.

To review the structure of a business object in Oracle HCM Cloud:

1. Open the **View Business Objects** page in the **Data Exchange** work area.
2. Search for the object by entering its name in the **Business Object** query-by-example field and pressing **Enter**.
3. Click the object name to open the Business Object Details page for the object. This page provides comprehensive information about the object's structure and attributes.

You need to define the transformation logic and build extraction routines. You can use tools that are native to the source system, such as PL/SQL in Oracle E-Business Suite or SQR in Oracle PeopleSoft. Alternatively, you can use an Extract, Transform, and Load tool, such as Oracle Data Integrator or PowerCenter Informatica.

**Validating the Source Data Before Upload:** Run the Data File Validator tool, available from My Oracle Support, in your source environment to test your generated .dat files. The utility generates a list of validation errors in HTML format. You can correct the errors in the .dat files before you load them.

## Unchanged Attributes:

When you update existing data in Oracle HCM Cloud, you need to supply: A unique identifier for the record being updated and the attributes that have changed.

Any attributes that you omit retain their current values. For performance reasons, you're recommended not to include attributes that haven't changed.

**Note:** For date-effective records, all attributes that you supply are updated for the specified date range.

**Null Attribute Values:** To set an attribute value explicitly to null, you must supply the #NULL token as the attribute value. You can't simply leave the attribute blank.

**Lookup-Validated Attributes:** or non-flexfield attributes that are defined in Oracle HCM Cloud as lookup types, you can specify either the lookup code or its meaning. For example, you can specify the sex of a person using either the lookup codes or the lookup meanings shown in this table.

Lookup Code	Lookup Meaning
M	Male
F	Female

You're recommended to use the lookup code, as the lookup meaning can be translated. The language of the lookup meaning must match the language of the user who's uploading the data.

Different rules apply to flexfield attributes that are defined as lookup types.

**Note:** You're recommended to use the lookup code, as you must supply the lookup meaning in the language configured for the FUSION\_APPS\_HCM\_ESS\_LOADER\_APPID elevated user, which may differ from language of the session user.

**Numeric Values:** For numeric values, only the decimal separator is supported. Don't include currency symbols, scientific notation, or thousands separators. To set an existing numeric value to null, supply the #NULL token as the attribute value.

### Date and Time Attributes

## How do I load images, attachments, and large strings?

You can load both Character Large Objects (CLOB) and Binary Large Objects (BLOB) using HCM Data Loader. However, the way in which you provide data for these objects is unique to these attributes.

Instead of supplying the data directly in the data (.dat) file, you supply it in a separate file. You supply the name of this file as the value of the relevant attribute in the data file.

This example shows a data file for the Document Attachment component. The **File** attribute references a text attachment file in each MERGE line.

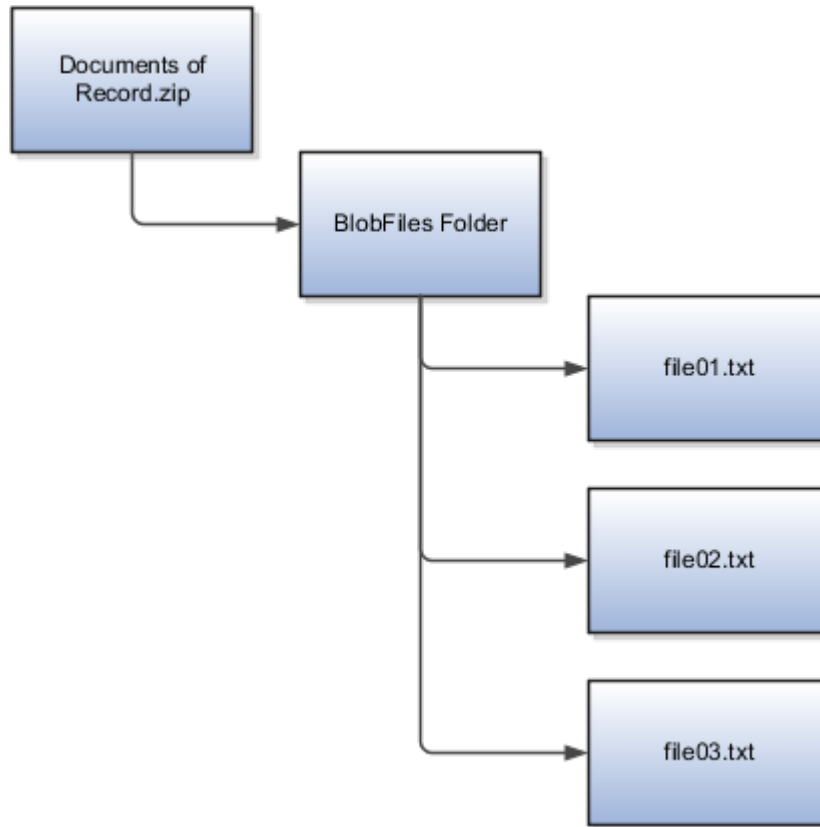
```
METADATA | DocumentAttachment | DocumentType | File | PersonNumber | ...
MERGE | DocumentAttachment | Drivers License | file01.txt | 23901 | ...
MERGE | DocumentAttachment | Drivers License | file02.txt | 64235 | ...
```

This approach is used because data for these data types can be very large. Also, data to be loaded directly rather than by attachment may need new-line characters, making it complex to include in the business-object data file.

### How to Supply CLOB and BLOB Files

To load data into a CLOB attribute, you supply the data in a separate file. You place the file in a **ClobFiles** folder in the same .zip file as the business-object data file. Similarly, to load data or upload an attachment to a BLOB attribute, you supply the data or file to attach in a **BlobFiles** folder. The data type of the attribute that's used to load a large object or attachment determines which folder to use.

The business-object documentation specifies the data types of all attributes. For example, the File attribute of the Document Attachment component has the BLOB data type. Therefore, you place referenced files in the BlobFiles folder,



as shown in this figure.

File names in the **ClobFiles** and **BlobFiles** folders can include UTF-8 single-byte characters. For example, file names can include the alphanumeric characters a through z, A through Z, and 0 through 9, underscore (\_), hyphen (-) and parentheses (.). Most file-name extensions are supported, though CLOB data is usually supplied in text (.txt) files.

Here's a tutorial on loading images and attachments using HDL: [Loading Images with Attachments with HCM Data Loader \(HDL\)](#)

## How do I include source keys in my data file?

The source key is made up of two attributes, SourceSystemOwner and SourceSystemId. If you use a source key to identify a record being merged, then you can also use source keys to identify integration-enabled foreign objects.

For example, you can identify the parent record of a child component using a source key. You can't use source keys for foreign-object references if you aren't also using a source key for the local record.

This topic describes how to supply source keys for both local records and foreign-object references. It also describes how default source keys are constructed.

## Defining Source-System Owner Values

The SourceSystemOwner attribute is common to all source keys supplied in a record. Therefore, all foreign objects that you identify using source keys must have the same SourceSystemOwner value as the record being merged. You must

define the SourceSystemOwner value by updating the HRC\_SOURCE\_SYSTEM\_OWNER lookup type before you use source keys.

## Specifying Source Keys for Local Records

To identify a local record uniquely using a source key, you supply values for both the SourceSystemId and SourceSystemOwner source-key attributes. This example shows how to identify a job object using a source key.

```
METADATA | Job | SourceSystemId | SourceSystemOwner | JobCode | JobName | SetCode | EffectiveStartDate | EffectiveEndDate  
MERGE | Job | 12349 | EBS-UK | SE | Software Engineer | COMMON | 2010/01/01 | 4712/12/31
```

## Specifying Source Keys for Foreign-Object References

To use a source key for a foreign-object reference, you append the hint (SourceSystemId) to the surrogate ID attribute for the foreign object. This example shows how to identify a Job object in an assignment record using a source key. JobId is the surrogate ID attribute for the Job object, to which you append the hint (SourceSystemId) in the METADATA line. You must have created the Job object using HCM Data Loader with the supplied source key.

```
METADATA | Assignment | SourceSystemId | SourceSystemOwner | JobId (SourceSystemId) | EffectiveStartDate |  
EffectiveEndDate  
MERGE | Assignment | 234234 | EBS-UK | 12349 | 2013/01/01 | 4712/12/31
```

**Note:** You can use source keys for foreign objects only if those objects are integration-enabled. The integrated business-object documentation identifies which objects are integration enabled.

If your foreign-object has a different source system owner for the local record, you must also supply the source system owner of the foreign-object reference. To do this, you append the hint (SourceSystemOwner) to the surrogate ID attribute for the foreign object. This example extends the example above and shows how to identify a Job object in an assignment record, where the source system owner for the Job and Assignment are different.

```
METADATA | Assignment | SourceSystemId | SourceSystemOwner | JobId (SourceSystemId) | JobId (SourceSystemOwner) |  
EffectiveStartDate | EffectiveEndDate MERGE | Assignment | 234234 | EBS-UK | 12349 | VISION | 2013/01/01 | 4712/12/31
```

## Default Source Keys

If you specify no source key when you create a record using HCM Data Loader, then a default source key is generated. The default SourceSystemOwner value is FUSION and the SourceSystemId is the surrogate ID. You can use source keys to identify records with default source keys. You can also update existing source keys, including default source keys, using the Source Key object. However, no process exists for extracting source-key information.

## How do I update source key associated with any record?

When you load data using HCM Data Loader, you can provide a source key. The source key is a value, usually generated from a legacy environment, that identifies the record uniquely in that environment.

You can use the source key to refer to the object both when maintaining it and when referring to it from another object. If you specify no source key, then a default source key is generated. You can update both default and locally defined source keys for integration-enabled objects.

## Loading Updates to Source Keys

To update the source key associated with any record, you load a SourceKey.dat file. In the file, you supply both a reference to the record to update and the new source-key value.

This example SourceKey.dat file updates the source key for a person address that's identified by its existing source key. The **BusinessObject** and **Component** attributes are used to identify the type of record being updated. For both attribute values, you supply the relevant file discriminator.

```
METADATA | SourceKey | BusinessObject | Component | OldSourceSystemId | OldSourceSystemOwner | NewSourceSystemId |  
NewSourceSystemOwner  
MERGE | SourceKey | Worker | PersonAddress | 2342 | FUSION | 1422-HOME | VISION
```

## How do I include user keys in my data file?

User keys are specific to the business object component that you're loading. The Business Object Details page in the View Business Objects task identifies the user-key attributes available for all business-object components and foreign-object references.

The user-key attributes are required when you create a record. They're also required when you update it, unless you supply a different key type to identify the record uniquely. This topic describes how to use user keys to identify local records and refer to foreign objects. It also describes how changes to user-key values affect their usefulness as keys.

## Specifying User Keys for Local Records

A user key can be made up of multiple attributes. You must supply them all if you're not using another key type, such as a source key, to identify the record. This example shows how to identify a Job object by its user key, which is made up of the JobCode and SetCode attributes.

```
METADATA | Job | JobCode | JobName | SetCode | EffectiveStartDate | EffectiveEndDate  
MERGE | Job | SE | Software Engineer | COMMON | 2010/01/01 | 4712/12/31
```

## Specifying User Keys for Foreign-Object References

In this example, an Assignment object is uniquely identified by its source key. However, a user key is used for the foreign-object reference to the associated Job object.

```
METADATA | Assignment | SourceSystemId | SourceSystemOwner | JobCode | SetCode | EffectiveStartDate | EffectiveEndDate  
MERGE | Assignment | 234234 | EBS-UK | SE | COMMON | 2013/01/01 | 4712/12/31
```

## Managing Changed User-Key Values

The values of some user keys aren't fixed. For example, you can change the name of an organization or location. As user keys can change, using them for historical references is difficult. If you're loading date-effective history for a business-object component where the user key has changed, then you must also supply a source key. This approach enables HCM Data Loader to identify related date-effective records correctly to form the object that you're loading.

## How do I include Oracle Fusion Surrogate IDs in my data file?

An Oracle Fusion surrogate ID is assigned automatically to a new record when you save it to the database. Surrogate IDs are specific to the business-object component that you're loading.

The integrated business-object documentation identifies the surrogate ID attribute for all business-object components and foreign-object references. This topic describes how to use surrogate IDs to identify local records and provide foreign-object references.

**Note:** Oracle HCM Cloud customers are unlikely to have access to the Oracle Fusion surrogate ID.

### Supplying Surrogate IDs for Local Records

This example shows how to use a surrogate ID to identify a Job component. For this component, the **JobId** attribute is its surrogate ID.

```
METADATA | Job | JobId | JobName | EffectiveStartDate | EffectiveEndDate  
MERGE | Job | 13413 | Software Engineer - Java | 2013/01/01 | 4712/12/31
```

### Supplying Surrogate IDs for Foreign-Object References

This example shows an Assignment component that's uniquely identified using a source key. The record includes a foreign-object reference to the associated Job object, which is identified by its surrogate ID, JobId.

```
METADATA | Assignment | SourceSystemId | SourceSystemOwner | JobId | EffectiveStartDate | EffectiveEndDate  
MERGE | Assignment | 234234 | EBS-UK | 13413 | 2013/01/01 | 4712/12/31
```

## How do I include Oracle Fusion GUIDs in my data file?

The Oracle Fusion GUID (Globally Unique Identifier) is a hexadecimal value that's assigned automatically to a record when it's saved to the database. This topic shows how to use GUIDs to identify local records and provide foreign-object references.

### Supplying GUIDs for Local Records

When you supply a GUID value to identify the record being merged or deleted, you use the attribute name GUID, regardless of the business-object component. This example shows how to specify a GUID value to identify a Job component.

```
METADATA | Job | GUID | JobName | EffectiveStartDate | EffectiveEndDate  
MERGE | Job | 2342UJFHI2323 | Software Engineer - Java | 2013/01/01 | 4712/12/31
```

## Supplying GUIDs for Foreign-Object References

To use a GUID for a foreign-object reference, you append the hint (GUID) to the surrogate ID attribute for the object to which you're referring. This example shows an Assignment component that's identified using source keys. The record includes a foreign-object reference to the associated Job object, which is identified by its GUID. JobId is the surrogate ID attribute for the Job object.

```
METADATA | Assignment | SourceSystemId | SourceSystemOwner | JobId (GUID) | EffectiveStartDate | EffectiveEndDate  
MERGE | Assignment | 234234 | EBS-UK | 2342UJHFI2323 | 2013/01/01 | 4712/12/31
```

You can use GUIDs for foreign objects only if they're integration enabled. The integrated business-object documentation identifies which foreign objects are integration enabled.

## How do I include reserved characters in my data file?

A few characters are reserved in the context of an HCM Data Loader data file. That is, they have a specific meaning by default, and you can't include them in attribute values unless you identify them explicitly as data.

This topic explains how to use the reserved characters in data that you're uploading. It also explains how to override the default reserved characters.

## Using Reserved Characters in Attribute Values

By default, these characters are reserved:

- Delimiter (vertical bar |)
- Newline character (n)
- Escape (backslash \)

To include the newline (n) and vertical bar (|) characters in attribute values, you precede them immediately with the escape character (backslash \). For example:

```
METADATA | Address | AddressLine1  
MERGE | Address | The Stables\|Main Allan
```

This entry enables the vertical bar to appear in the attribute value for address line 1:

The Stables|Main Allan

To include the newline character in a value, you specify \n. For example:

```
METADATA | Address | AddressLine1  
MERGE | Address | The Stables\nMain Allan
```

This entry results in the following value for address line 1:

The Stables

Main Allan

## Overriding the Reserved Characters

You can override the reserved characters for a file using the SET file-line instruction, which must appear before any METADATA lines in the file. The SET commands for overriding reserved characters have this format:

```
SET FILE_ESCAPE <new_value>
SET FILE_DELIMITER <new_value>
SET FILE_NEW_LINE <new_value>
```

The new value can be up to 10 characters. For example, you could set the newline character to newline and the file delimiter to comma (,) using the following SET commands:

```
SET FILE_DELIMITER ,
SET FILE_NEW_LINE newline
```

In this case, METADATA and MERGE lines could appear as follows:

```
METADATA,Address,AddressLine1
MERGE,Address,TheSteading\newlineKier Allan
```

**Tip:** You can also override the reserved characters for the enterprise by setting relevant configuration parameters. For example, set the **File Delimiter** configuration parameter to override the default file delimiter for the enterprise.

## How do I load and maintain translated business objects?

In environments where multiple languages are enabled, you can use HCM Data Loader to upload translated objects.

You specify the character set of the data file by naming any Java-supported character set on the **File Character Set** configuration parameter. The default character set is UTF-8.

### Loading Translated Objects

Loading translated objects is a two-stage process:

1. You create the object by loading the base-language version. At this stage, translation records are created for all enabled languages, but they hold base-language versions of any translatable values. For example, if US English is your base language, then translation records hold US English versions of translatable values.
2. You load the translated values as corrections to the base-language object. To perform this step, you use data-file templates that are provided specifically for translation. One translation data-file template is provided for each business-object component that includes translatable values.

For example, you may create the Sales Manager job in an environment where US English is the base language. If French, German, and Spanish are also enabled, then the object is created as shown in this table.

Language	Source Language	Job Name
US English	US English	Sales Manager
French	US English	Sales Manager

Language	Source Language	Job Name
German	US English	Sales Manager
Spanish	US English	Sales Manager

Once this object exists, you can load a single translation data file (JobTranslation.dat) to correct the French, German, and Spanish versions of the job name. Alternatively, you can load a translation file for each language if you prefer. You can deliver translation files either in the same .zip file as the original object or separately. However, you can't deliver them before the base-language object exists.

## Updating Translated Data

To update existing translated data, you can upload just the relevant translation file, provided that the object isn't date effective.

You must also update the base-language object if:

- The object is date effective.
- The object currently has no date-effective record with the same effective start date as the new translation value.

This requirement exists to ensure that the effective dates of the base-language and translation objects remain the same.

You can't delete translation objects in isolation. Translation objects are deleted automatically when you delete the associated object. For example, when you delete a Job Family object any associated translation objects are deleted automatically.

## Translation-File Discriminators

Unique file discriminators exist for the translation files and are identified in relevant files. For example, the file discriminator for the file JobTranslation.dat is JobTranslation.

This example shows a Job.dat file followed by the associated JobTranslation.dat file.

```
METADATA | Job | SourceSystemOwner | SourceSystemId | EffectiveStartDate | EffectiveEndDate | JobCode | Name | ActiveStatus
MERGE | Job | EBS-UK | JB2ACC44 | 2010/01/01 | 2014/04/04 | ACADM | Accounts Administrator | A
MERGE | Job | EBS-UK | JB2ACC44 | 2014/04/05 | 4712/12/31 | ACADM | Accounts Clerk | A

METADATA | JobTranslation | SourceSystemOwner | SourceSystemId | EffectiveStartDate | EffectiveEndDate | Language | Name
MERGE | JobTranslation | EBS-UK | JB2ACC44 | 2010/01/01 | 2014/04/04 | ES | Administrador de Cuentas
MERGE | JobTranslation | EBS-UK | JB2ACC44 | 2014/04/05 | 4712/12/31 | ES | Cuentas Clerk
```

## How do I include reference values in data files?

You can include reference values in each data line in a file.

They enable you to record reference data for the file line, which could include values already supplied on the file line or within the object, or source-system information, such as the database-table name, column names, and attribute

values from the source system. The Object Status page within the Import and Load Data interface, displays the reference values supplied against the top-level record in your object. Navigate to Object Details page to view the reference information supplied against each file line that constructs your object. You can create OTBI reports to include these values using the HCM Integrations Real Time subject area. The reference names and values are available in the Imported Objects sub-folder within the HCM Data Loader folder.

For example, when loading element entries, you may want to filter and reconcile by the assignment number, element name and amount. You can supply these values against the ElementEntry record, even though the amount is loaded using the ElementEntryValue record type. This would then allow you to filter by the assignment number and element name, to check the status of the element entry, while reviewing the amount specified for the element entry.

You aren't restricted as to the values you supply, there is no validation of these attributes. The data supplied to them in held only the staging tables, it is not saved to the application tables.

References are optional. This topic describes how to construct reference attributes, which are made up of a name and a value.

## Reference Names

You specify reference context and column names in the METADATA line. To specify a source-system database-table name or reference context, you add the following entry to the METADATA line:

```
SourceRefTableName=<table name>
```

You can specify up to 10 reference column names in the same METADATA line using the SourceRef001 to SourceRef010 tags.

Examples:

To supply details of the source record on your legacy system:

```
METADATA | Job | SourceRefTableName=PER_JOBS | SourceRef001=JOB_ID | SourceRef002=EFFECTIVE_START_DATE |  
SourceRef003=EFFECTIVE_END_DATE
```

To supply reconciliation and reporting information:

```
METADATA | ElementEntry | SourceRef001=Assignment Number | SourceRef002=Element Name | SourceRef003=Effective Start  
Date | SourceRef004=Amount
```

Reference attributes can appear anywhere in the METADATA line after the instruction and discriminator values.

## Reference Values

Supply the reference values on each data line, ensuring that they appear in the order specified on the METADATA line. In data lines, you must leave the SourceRefTableName attribute blank. This value appears in the METADATA definition only. For example:

```
METADATA | Job | SourceRefTableName=PER_JOBS | SourceRef001=JOB_ID | SourceRef002=EFFECTIVE_START_DATE |  
SourceRef003=EFFECTIVE_END_DATE  
MERGE | Job | | 135 | 2010/01/01 | 4712/12/31  
MERGE | Job | | 136 | 2010/01/01 | 4712/12/31
```

## How do I delete a business object or its components?

You can delete many objects using HCM Data Loader, regardless of whether they were loaded using HCM Data Loader. This topic explains how to delete an object or its components and identifies any restrictions.

**Tip:** As you can't recover a deleted object, try to correct your data instead of deleting and recreating it.

### What You Can and Can't Delete

You can delete:

- Most complete business objects, except Worker.
- Most individual business-object components.
- Some child components of the Worker object. For example, you can delete the Person Email component of a worker object.

When you delete a parent object, its child components and any translation objects are also deleted. For example, to delete a Grade object and its child components, you create a DELETE instruction for the Grade discriminator. To delete only a Grade Rate Value child component, you create a DELETE instruction for the GradeRateValue discriminator.

You can't delete:

- Individual date-effective records
- Individual translation objects
- Worker objects

**Note:** Before you delete an object, ensure that other business objects don't refer to it.

For information about whether you can delete a component, see the business-object details on the View Business Objects page in the Data Exchange work area.

### DELETE Instructions

To delete an object, include the DELETE instruction in the relevant data file. For example, you could delete a Job Family object by including these lines in the JobFamily.dat file:

```
METADATA|JobFamily|EffectiveStartDate|EffectiveEndDate|JobFamilyName  
DELETE|JobFamily|2012/10/01|4712/12/31|Sales01
```

These rules apply:

- You can't include DELETE instructions in translation data files.
- You must not supply a DELETE instruction for a record that has a MERGE instruction in the same file. HCM Data Loader doesn't know which instruction to process first.

## Deleting Date-Effective Objects

To delete a date-effective object identified by user keys, you must supply both the effective start and effective end dates. If the object is identified using any of the other key types, then the effective start and end dates are optional.

## What are the considerations for loading date-effective objects?

Many Oracle HCM Cloud objects are date-effective. That is, they retain a history of changes, each of which has effective start and end dates. Professional users can retrieve and edit the version of an object as of a specified date.

You can load date-effective objects and their date-effective history using HCM Data Loader. This topic provides some general rules about loading date-effective objects.

## Attributes of Date-Effective Records

This table introduces the attributes that occur in date-effective records.

Attribute	Description
<b>EffectiveStartDate</b>	The start date for the attribute values. This value is required for all date-effective records.
<b>EffectiveEndDate</b>	The end date for the attribute values. If you leave this value blank, then the date-effective record continues by default to the end of time.
<b>EffectiveSequence</b>	When multiple changes per day occur, this attribute identifies their order.
<b>EffectiveLatestChange</b>	For objects with multiple changes per day, this attribute identifies the latest record for the effective start date.

## Supplying Date-Effective History

You can decide how much history to load for new objects, but the history that you provide must be complete and valid. Although you can supply date-effective records in any order in the .dat file, no break in the dates can occur. When you create a date-effective object, only the first date-effective record is processed as a new record. Later-dated records are updates to the first record. To retain values from the previous date-effective record, you can either leave those attribute values blank or repeat the value that's to continue.

**Note:** Leaving an attribute value blank doesn't set the attribute to null. To change a value to null, you must supply the #NULL token as the attribute value.

## Key Values

HCM Data Loader groups records into logical objects, where a logical object is one occurrence of the business object, such as a Worker. The records belonging to a logical object are identified by the unique key for the component. Therefore, the key value must be the same throughout the date-effective history. You can use any of the four key types.

## Creating Date-Effective Objects

When you create business objects containing date-effective components using HCM Data Loader:

- Include the `SET PURGE_FUTURE_CHANGES Y` instruction at the start of the data file. This instruction sets the HCM Data Loader date-effective maintenance mode to Replace. In this mode, which is the default, the contents of the data file replace any existing data.
- If you're including object history, then supply the `#NULL` token for all attributes that must have null values.

## Updating Date-Effective Objects

When you load updates to objects that contain date-effective components:

- Include the `SET PURGE_FUTURE_CHANGES N` instruction at the start of the data file. This instruction sets the HCM Data Loader date-effective maintenance mode to Retain. In this mode, existing future-dated changes are retained. If you don't include this instruction, then any existing future-dated changes for the date range of your updates are purged.
- Always supply the effective start date of each change. The effective end date is optional. However, you must set the effective end date to `#RETAIN` if you don't want to correct any future-dated records. Leave the effective end date null if you want a change to apply until the end of time.

**Note:** When you update date-effective objects, the values that you supply are applied to every record in the specified date-range. However, some attributes, such as **ActionCode**, are protected from update.

## How do I load objects with multiple changes per day?

For most date-effective objects, only one record exists for a specific date. Therefore, the object history shows only the last update to each attribute on any day.

However, for some objects, such as Assignment, a record exists for each update in a day. Such objects include an effective sequence number to show the update order. Changes are cumulative, and the record with the highest effective sequence number is available to transactions for that date. This topic describes how to load and maintain objects that support multiple changes per day.

## Effective Sequence Number

When you load date-effective history for an object with multiple changes on a single date, the **EffectiveSequence** value must start at 1 and increase sequentially. You can't repeat an **EffectiveSequence** value for the same logical object on the same date, nor can gaps exist in the sequence. If only one change exists for an effective start date, then **EffectiveSequence** must be 1.

**Note:** Don't leave **EffectiveSequence** blank when you're loading multiple changes for a single day. Without this information, the order in which records with the same effective start date are processed is undefined.

## Effective Latest Change

When you report multiple changes on the same effective start date, the **EffectiveLatestChange** attribute of the last record in the sequence must be **Y**. For all earlier records, the attribute must be **N**. **EffectiveLatestChange** is a required attribute for records that support multiple changes per day. If only one change exists on an effective start date, then **EffectiveLatestChange** must be **Y**.

## Effective Start and End Dates

The effective end date of any record with **EffectiveLatestChange** set to **N** must be the same as its effective start date.

## Loading Date-Effective History with Multiple Changes Per Day: Example

This example loads date-effective history for an Assignment object.

```
METADATA|Assignment|SourceSystemId|EffectiveStartDate|EffectiveSequence|EffectiveLatestChange|EffectiveEndDate|...
MERGE|Assignment|2724|2010/06/08|1|Y|2012/03/03|...
MERGE|Assignment|2724|2012/03/04|1|N|2012/03/04|...
MERGE|Assignment|2724|2012/03/04|2|N|2012/03/04|...
MERGE|Assignment|2724|2012/03/04|3|Y|2012/06/01|...
MERGE|Assignment|2724|2012/06/02|1|Y||...
```

This example observes these rules:

- No gaps exist in the dates.
- The unique key is identical for all date-effective records.
- **EffectiveSequence** starts at **1** for all new effective start dates and is sequential for matching effective start dates.
- **EffectiveLatestChange** is **Y** when the record is the last change on an effective start date. Otherwise, **EffectiveLatestChange** is **N**.
- The **EffectiveEndDate** matches the **EffectiveStartDate** when **EffectiveLatestChange** is **N**.

## Updating Objects with Multiple Changes Per Day

When supplying a single update to an existing object that support multiple changes per day, you may not know the next available sequence number. In this case, leave the **EffectiveSequence** attribute blank. The next number in the existing sequence is generated automatically.

To correct an existing record, you must supply the existing values for these attributes to identify the record correctly:

- Effective Start Date
- Effective End Date
- Effective Sequence
- Effective Latest Change

For example:

```
METADATA | Assignment | SourceSystemId | EffectiveStartDate | EffectiveSequence | EffectiveLatestChange |
EffectiveEndDate | ...
MERGE | Assignment | 2724 | 2010/06/08 | 3 | N | 4712/12/31 | ...
```

**Note:** Don't leave **EffectiveSequence** blank when you're loading multiple changes for a single day. Without this information, the order in which records with the same effective start date are processed is undefined. However to leave the **EffectiveSequence** blank, ensure that you include the SET PURGE\_FUTURE\_CHANGES N instruction in the data file. For more information on this instruction, refer Example of Retaining Attribute Values in Future-Dated Records. Additionally, for details on supplying values when updating Date-effective objects, refer Updating Date-Effective Objects.

## How are updates to future-dated records processed in Retain and Replace maintenance modes?

When you update a date-effective object, your changes may affect more than one of its existing date-effective records. Also, some of the affected records may have effective start dates that are later than the effective start date of the update.

This table shows the date-effective history of a Job object.

Effective Start Date	Effective End Date	Job Code	Job Name	Regular Temporary	Full Part Time	Active Status
2010/06/08	2012/01/09	ACC1	Accounts Clerk	T	PART_TIME	A
2012/01/10	2012/03/03	ACC1	Accounts Clerk	T	FULL_TIME	A
2012/03/04	4712/12/31	ACC1	Accounts Administrator	T	FULL_TIME	A

If you update the object specifying an effective start date of 2011/01/01, then a new date-effective record is generated for 2011/01/01. This update changes the effective end date of the 2010/06/08 record. It also affects both the 2012/01/10 and 2012/03/04 records. The exact effect on these existing future-dated records depends on:

- The maintenance mode, which can be either Retain or Replace
- The effective end date value, if any, that you supply

### Retain Mode or Replace Mode

This table summarizes the combined effects of the maintenance mode and effective end date on existing future-dated records.

Mode	Effective End Date Value	Description
Retain	#RETAIN	Existing attribute values in future-dated records are retained. Your updates don't affect these values. All existing records are retained.
Retain	#ALL	Changed attribute values are carried forward to all future-dated records until the end of time. All existing records are retained.  <b>Tip:</b> Leaving the effective end date blank or setting it to 4712/12/31 has the same effect as specifying the #ALL token. However, you must use the #ALL token to carry forward changes for objects that have a specific end date.
Retain	Specific date other than 4712/12/31	Changed attribute values are carried forward to future-dated records until the specified end date. All existing records are retained.
Replace	Specific date or blank	A single record replaces existing date-effective records for the date range specified on the update.

Retain mode is recommended when you're applying an incremental update to an existing record. Replace mode, which is the default mode, is useful during data migration, when you're uploading the complete data for a record.

**Note:** Replace mode may have unexpected consequences if you upload less than the complete data for a record.

In both modes:

- A new date-effective record is generated for the object if you specify an effective start or end date for which no date-effective record currently exists.
- Only the attributes for which you supply values are updated.

## How do I set the Retain and Replace maintenance modes?

Use the SET command to set the maintenance mode for a date-effective object. This table shows the set command to include in the data file.

Mode	SET Command
Retain	SET PURGE_FUTURE_CHANGES N
Replace	SET PURGE_FUTURE_CHANGES Y

If you include neither of these SET commands in a data file, then the default mode is Replace. The default mode can be overridden for the enterprise using the **Date-Effective Update Mode** configuration parameter.

## How are future-dated records preserved when using Retain mode?

When you update a date-effective object, you can leave attribute values in future-dated records unchanged. To retain these values, you set the maintenance mode to Retain and specify the #RETAIN token instead of an effective end date.

This token ensures that your changes take effect from the supplied effective start date to the start of the next record. If no future-dated records exist, then your changes apply until the end date of the record. This topic provides an example showing the effects of Retain mode.

### Retaining Future-Dated Attribute Values in an Assignment Object

Consider the date-effective history of the Assignment object shown in this table.

Effective Start Date	Effective Sequence	Effective End Date	Action Code	Job Code	Grade	Location	Hours
2010/06/08	1	2012/03/03	HIRE	ACC1	IC2	HQ	40
2012/03/04	1	2012/03/04	PROMOTION	ACC1	IC4	HQ	40
2012/03/04	2	2012/03/04	TRANSFER	ACC1	IC4	LVP	40
2012/03/04	3	2012/06/01	JOB_CHANGE	ACC3	IC4	LVP	40
2012/06/02	1	4712/12/31	JOB_CHANGE	ACC2	IC4	LVP	40

To change the working hours to 37.5 from 2012/01/10 without affecting future-dated records, you use this data file:

```
SET PURGE_FUTURE_CHANGES N
METADATA | Assignment | SourceSystemId | EffectiveStartDate | EffectiveSequence | EffectiveLatestChange |
EffectiveEndDate | ActionCode | NormalHours
MERGE | Assignment | 2724 | 2012/01/10 | Y | #RETAIN | ASG_CHANGE | 37.5
```

After update, the Assignment object is as shown in this table.

Effective Start Date	Effective Sequence	Effective End Date	Action Code	Job Code	Grade	Location	Hours
2010/06/08	1	2012/01/09	HIRE	ACC1	IC2	HQ	40
2012/01/10	1	2012/03/03	ASG_CHANGE	ACC1	IC2	HQ	37.5
2012/03/04	1	2012/03/04	PROMOTION	ACC1	IC4	HQ	40
2012/03/04	2	2012/03/04	TRANSFER	ACC1	IC4	LVP	40
2012/03/04	3	2012/06/01	JOB_CHANGE	ACC3	IC4	LVP	40
2012/06/02	1	4712/12/31	JOB_CHANGE	ACC2	IC4	LVP	40

Attributes in records with effective start dates after 2012/01/10 retain their current values.

**Note:** As recommended, the data file includes no effective sequence number. Therefore, the next available number in the sequence is allocated automatically.

## What are the options to carry forward changed attribute values in Retain mode?

When you update a date-effective object in Retain mode, you may want to carry changes forward to future-dated records. In this case, you can either specify an effective end date or carry the changes forward to all existing future-dated records.

### Specifying an Effective End Date

This table shows the date-effective history of a Job object.

Effective Start Date	Effective End Date	Job Code	Job Name	Regular Temporary	Full Part Time	Active Status
2010/06/08	2012/01/09	ACC1	Accounts Clerk	T	PART_TIME	A
2012/01/10	2012/03/03	ACC1	Accounts Clerk	T	FULL_TIME	A
2012/03/04	4712/12/31	ACC1	Accounts Administrator	T	FULL_TIME	A

To change the **Regular Temporary** value from **T** to **R** for the period from 2011/03/04 to 2014/04/04, you include these instructions in the .dat file:

```
SET PURGE_FUTURE_CHANGES N
METADATA | Job | SourceSystemId | EffectiveStartDate | EffectiveEndDate | RegularTemporary
MERGE | Job | 45346 | 2011/03/04 | 2014/04/04 | R
```

After update, the Job object is as shown in this table.

Effective Start Date	Effective End Date	Job Code	Job Name	Regular Temporary	Full Part Time	Active Status
2010/06/08	2011/03/03	ACC1	Accounts Clerk	T	PART_TIME	A
2011/03/04	2012/01/09	ACC1	Accounts Clerk	R	PART_TIME	A
2012/01/10	2012/03/03	ACC1	Accounts Clerk	R	FULL_TIME	A
2012/03/04	2014/04/04	ACC1	Accounts Administrator	R	FULL_TIME	A
2014/04/05	4712/12/31	ACC1	Accounts Administrator	T	FULL_TIME	A

A new date-effective record is created for both the start and end dates of the change. The **Regular Temporary** attribute is updated for the specified period. All other attributes keep their existing values.

## Applying Changes to All Future-Dated Records

You can use the #ALL token in place of an effective end date. This token specifies explicitly that your changes apply to all future-dated records, regardless of the object's end date.

For example, you want to make these changes to the Job object:

- Change the **Regular Temporary** value from **T** to **R**.
- Set the effective start date of the change to 2011/03/05.
- Apply the changes to all future-dated records.

You include these instructions in the data file:

```
SET PURGE_FUTURE_CHANGES N
METADATA | Job | SourceSystemId | EffectiveStartDate | EffectiveEndDate | RegularTemporary
MERGE | Job | 45346 | 2011/03/04 | #ALL | R
```

After update, the Job object is as shown in this table.

Effective Start Date	Effective End Date	Job Code	Job Name	Regular Temporary	Full Part Time	Active Status
2010/06/08	2011/03/03	ACC1	Accounts Clerk	T	PART_TIME	A

Effective Start Date	Effective End Date	Job Code	Job Name	Regular Temporary	Full Part Time	Active Status
2011/03/04	2012/01/09	ACC1	Accounts Clerk	R	PART_TIME	A
2012/01/10	2012/03/03	ACC1	Accounts Clerk	R	FULL_TIME	A
2012/03/04	4712/12/31	ACC1	Accounts Administrator	R	FULL_TIME	A

You can achieve the same result by leaving the effective end date in the .dat file blank or setting it explicitly to 4712/12/31. However, you must use the #ALL tag for any object that has an explicit end-date.

## How are future-dated records handled in Replace mode when you replace attribute values?

When you update a date-effective object in Replace mode, you replace all existing, future-dated records with the uploaded data. You're recommended to use Replace mode only when the contents of the .dat file must replace the existing data rather than update it.

Replace mode is the default maintenance mode. However, you can use this SET command to select Replace mode explicitly and record the setting in the .dat file:

```
SET PURGE_FUTURE_CHANGES Y
```

## Updating Only Selected Attributes of an Object in Replace Mode

This table shows the date-effective history of an Assignment object.

Effective Start Date	Effective Sequence	Effective End Date	Action Code	Job	Grade	Location	Normal Hours
2010/06/08	1	2012/03/03	HIRE	ACC1	IC2	HQ	40
2012/03/04	1	2012/03/04	PROMOTION	ACC1	IC4	HQ	40
2012/03/04	2	2012/03/04	TRANSFER	ACC1	IC4	LVP	40
2012/03/04	3	2012/06/01	JOB_CHANGE	ACC3	IC4	LVP	40
2012/06/02	1	4712/12/31	JOB_CHANGE	ACC2	IC4	LVP	40

To update the working hours to 37.5 from 2012/01/10 in Replace mode, you create the following data file:

```
SET PURGE_FUTURE_CHANGES Y
```

```
METADATA|Assignment|SourceSystemId|EffectiveStartDate|EffectiveSequence|EffectiveLatestChange|
EffectiveEndDate|ActionCode|NormalHours
MERGE|Assignment|2724|2012/01/10|Y||ASG_CHANGE|37.5
```

After update, the Assignment object is as shown in this table.

Effective Start Date	Effective Sequence	Effective End Date	Action Code	Job	Grade	Location	Normal Hours
2010/06/08	1	2012/01/09	HIRE	ACC1	IC2	HQ	40
2012/01/10	1	4712/12/31	ASG_CHANGE	ACC1	IC2	HQ	37.5

In Replace mode, records with effective start dates that are later than the effective start date of the update are purged. Attribute values that predate the change and for which you supply no values in the data file appear in the new record. This update isn't reversible.

**CAUTION:** You're recommended to supply values for all attributes that currently exist on the object when updating in Replace mode. In this example, you could have provided values for job, grade, and location. As you didn't supply those values, the values that are current on the effective start date of the update remain in effect.

## How do I change the first effective start date or last effective end date for some business objects?

For some objects, you can change the first effective start date or last effective end date.

To change an object's first effective start date, you include the **ReplaceFirstEffectiveStartDate** attribute in the .dat file and set it to **Y**. To change an object's last effective end date, you include the **ReplaceLastEffectiveEndDate** attribute in the .dat file and set it to **Y**. You supply the revised dates on the **EffectiveStartDate** or **EffectiveEndDate** attribute, as appropriate. You can make these changes at the same time as other changes. For example, you can change both the start date of an object and its name.

**Note:** The **ReplaceFirstEffectiveStartDate** and **ReplaceLastEffectiveEndDate** parameters don't appear in object template files or on the Attributes tab of the Business Object Details page. However, you can add them to .dat files for date-effective objects that support these changes.

These restrictions apply to changing first effective start date and last effective end date:

- For some objects, the last effective end date must be 4712/12/31. Any attempt to change this date causes an error.
- You can't change the first effective start date of the Worker object in this way. Instead, you must use the **NewStartDate** attribute.
- When you include the **ReplaceLastEffectiveEndDate** attribute in a .dat file, you must specify a date value for the **EffectiveEndDate** attribute. You must not supply a #RETAIN, #ALL, or #NULL token in place of the effective end date.

**Tip:** On the Component Details tab of the Business Object Details page, identify whether you can change a component's first effective start or last effective end date. For example, in the Date Effectivity section on the Component Details tab for the Job object, the **First start date** option is selected. Therefore, you can include the **ReplaceFirstEffectiveStartDate** parameter in the .dat file.

## Changing the First Effective Start Date of a Job Object

This example shows how to include the **ReplaceFirstEffectiveStartDate** attribute in a .dat file.

```
METADATA | Job | JobId | EffectiveStartDate | EffectiveEndDate | ReplaceFirstEffectiveStartDate  
MERGE | Job | 23452 | 1950/01/01 | 4712/12/31 | Y
```

## Changing the Last Effective End Date of a Recurring Element Entry

This example shows how to include the **ReplaceLastEffectiveEndDate** attribute in a .dat file.

```
METADATA | ElementEntry | ElementEntryId | EffectiveStartDate | EffectiveEndDate | ReplaceLastEffectiveEndDate  
MERGE | ElementEntry | 4634 | 2014/01/01 | 2014/04/05 | Y
```

# What are the guidelines for loading descriptive and extensible flexfield data?

Using HCM Data Loader, you can load data for both descriptive flexfields and extensible flexfields. This topic explains concepts that are common to both types of flexfields.

## Configuring Flexfields

Before you generate template files for objects with flexfields, you must configure the flexfields in Oracle HCM Cloud. After the flexfields are configured, the METADATA lines in generated template files include all attributes that are required to load the flexfield data. You also see a Flexfield Attributes tab for relevant object components on the Business Object Details page in the Data Exchange work area.

## Flexfield Codes

When loading flexfield data, you must supply the flexfield code in the METADATA line in this format:

```
FLEX:<flexfield code>
```

For example, to load data for the job descriptive flexfield, the flexfield code is:

```
FLEX:PER_JOBS_DFF
```

Relevant flexfield codes appear in the generated template file and on the Flexfield Attributes tab for the business object.

## Flexfield Attributes

Flexfield attribute names are those that you specify when configuring the flexfield. Both descriptive and extensible flexfields have one or more contexts. When you include a flexfield attribute name on the METADATA line for an object, you must also identify the context. You provide a hint value in the following format to identify both the flexfield and the context to which the attribute belongs:

```
<flexfield attribute name>(<flexfield code>=<context code>)
```

For example, for the `_MAIN_CONTRACT` attribute of the `PER_CONTRACT_LEG_DDF` descriptive flexfield with the `CN` context code, the entry on the `METADATA` line is:

```
_MAIN_CONTRACT (PER_CONTRACT_LEG_DDF=CN)
```

## Business Objects with Multiple Descriptive Flexfields

Some business object components support multiple descriptive flexfields. You can include all descriptive flexfield attributes for every flexfield and configured context on a single `METADATA` line. This approach is possible because you supply both the flexfield code and the context for each attribute.

## Lookup and Value Set Validated Flexfield Values

For flexfield values that are validated by lookups, you can supply either the lookup code or the lookup value. To supply the value, you must append the suffix `_Display` to the attribute name in the `METADATA` line.

For example, for a configured gender flexfield attribute, you can supply either the lookup code (M or F) or the value (Male or Female). To supply the value, use the attribute name `gender_Display`, as shown in this example:

```
METADATA | Job | FLEX:PER_JOBS_DFF | gender_Display (PER_JOBS_DFF=JOBCONTEXT1) | ...  
MERGE | Job | JOBCONTEXT1 | Male | ...  
MERGE | Job | JOBCONTEXT1 | Female | ...
```

The same is true for flexfield segments validated by table-validated value sets. You can supply either the ID or value. To supply the value, you must append the suffix `_Display` to the attribute name in the `METADATA` line.

**Note:** You're recommended to use the lookup code or value set ID, as the lookup meaning and value set value can be translated. The language of the value provided to the `_Display` attribute must match the language of the user who's uploading the data. For HCM Data Loader that's the elevated `FUSION_APPS_HCM_ESS_LOADER_APPID` user, whose language may differ from language of the session user.

## Secured Value Sets

A value set is a group of values that validate the content of a flexfield segment. If the value set is security enabled, then it's available to specific users or processes only. You must ensure that the predefined `FUSION_APPS_HCM_ESS_LOADER_APPID` user has access to any security-enabled value sets that your flexfields reference. Otherwise, the load fails. For more information about enabling access to secured value sets, see the *Securing HCM* guide.

## Deleting Flexfield Segments

Let's say, for GDPR or other compliance purposes, you need to remove values from a 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.

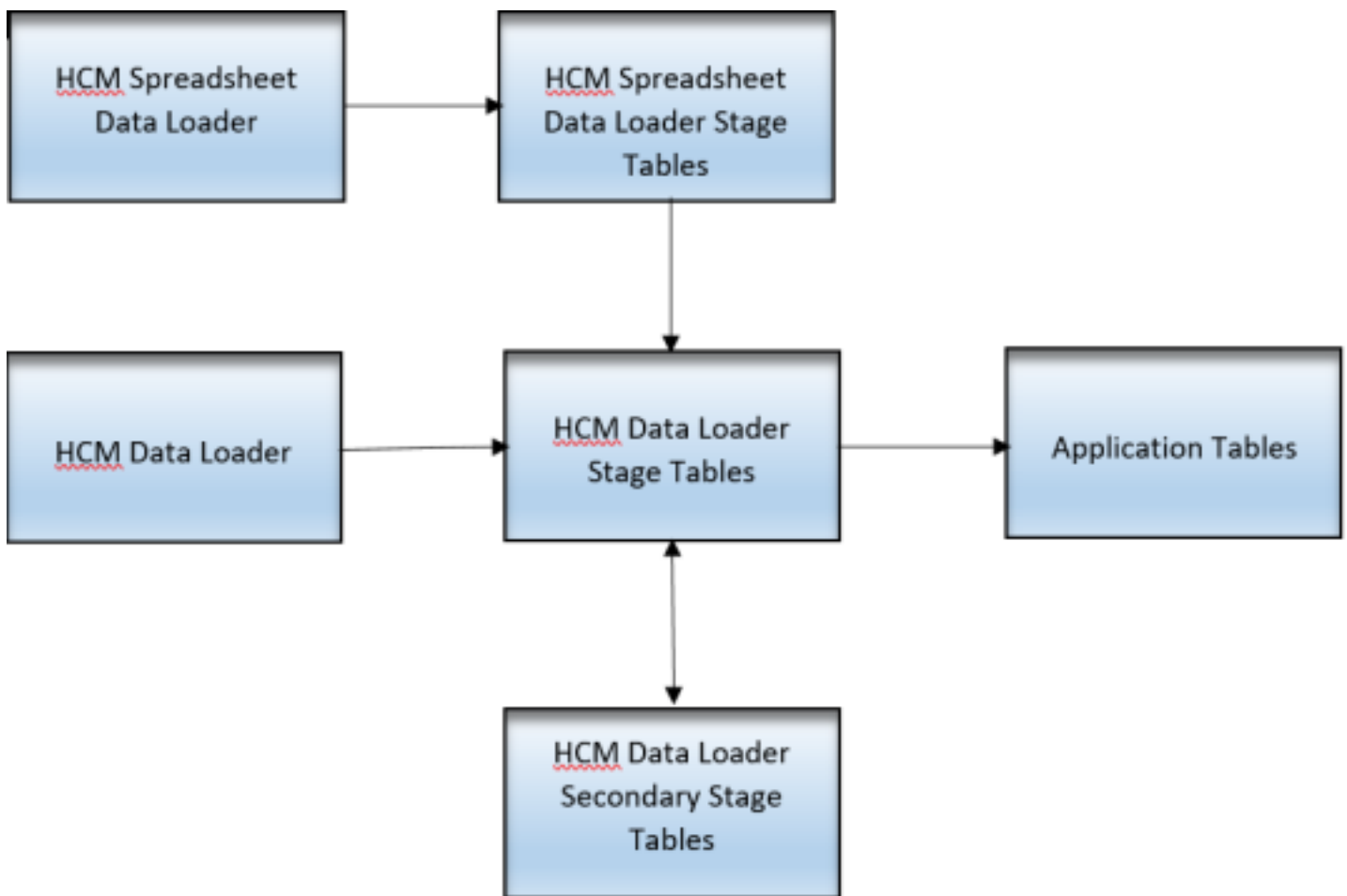
Here's a tutorial on loading flexfield segments using HDL: [Loading Flexfield Segments with HCM Data Loader \(HDL\)](#)

Here's a tutorial to add preconfigured and custom flexfield segments to your HCM Spreadsheet Data Loader templates [Including Flexfield Segments in HCM Spreadsheet Data Loader Templates](#)

## How is data loaded to Oracle HCM Cloud?

Data that you load using either HCM Data Loader or HCM Spreadsheet Data Loader is imported first to stage tables, where validation occurs. HCM Spreadsheet Data Loader has its own stage tables, from where valid data is loaded to the HCM Data Loader stage tables.

Valid data is loaded from the HCM Data Loader stage tables to the application tables. When a data set contains data for an object that supports roll back, such as element entry, the data for the entire data set can be moved to secondary stage tables, ensuring it's not purged prematurely and doesn't negatively impact the bulk load processing which occurs from the main stage tables



**Note:** In the Import and Load Data page, data sets that are retained for an extended period in the secondary stage table can be identified by the green tick that are retained for an extended period in the secondary stage table can be identified by the green tick in the Extended Retention column.

## How to extend the period of retention for a data set?

The Enable Automatic Extended Data Set Retention configuration parameter determines if data sets containing data that can be validated or rolled back are automatically moved to the secondary staging tables. The extended retention period is defined by the Days to Preserve Extended Retention Data Sets configuration parameter.

You can override the default behavior at the data set level by using the `EXTEND_DATA_SET_RETENTION` SET Instruction. If you have loaded a data set and it is held for the standard retention period within the primary staging tables, you can extend the data set retention by moving it to the secondary staging tables from the Import and Load Data page.

## How processed data sets are deleted from stage tables?

You can delete processed data sets from three sets of stage tables.

- To delete HCM Spreadsheet Data Loader data sets from all stage tables, use the **Delete Spreadsheet Stage Table Data** task. It runs the **Delete HCM Spreadsheet Data Loader Stage Table Data** process.
- To delete data sets from the HCM Data Loader extended retention stage tables, use the **Delete Stage Table Data** task. It runs the **Delete HCM Data Loader Stage Table Data** process.

Alternatively, you can delete individual data sets from these stage table using the Delete button on the Data Sets table within the Import and Load Data interface.

Perform all tasks in the Data Exchange work area.

You can delete:

- Individual data sets
- All data sets that currently match specified criteria

**Note:** You can schedule a recurring deletion request. For example, you can schedule weekly deletion of all data sets that haven't been updated in the last 7 days.

When you run the **Import and Load Data** process, it determines whether a schedule exists for the **Delete HCM Data Loader Stage Table Data** process. If no schedule exists, then the process creates one to run **Delete HCM Data Loader Stage Table Data** nightly.

**Note:** If the **Import and Load Data** process identifies that an existing schedule is failing because the initiating user is no longer having the permissions to run the process, the schedule will be automatically deleted with a warning.

When you delete extended retention data sets from the HCM Spreadsheet Data Loader stage tables, they're deleted automatically from the HCM Data Loader stage tables. You can also delete spreadsheet data sets explicitly from the HCM Data Loader stage tables. In this case, they remain available in the HCM Spreadsheet Data Loader stage tables. You can't recover deleted data sets. Don't delete data sets that you want to report on or that you want to extract error information for.

## How Delete HCM Data Loader Stage Table Data process deletes data sets?

The **Delete HCM Data Loader Stage Table Data** process deletes data sets from the HCM Data Loader and extended retention stage tables. You can delete from one set of stage tables at a time and this is determined by the data set type. Whenever the process runs for standard data sets, the process:

1. Purges automatically from the stage tables any data set that hasn't been updated in the last 30 days. You can specify a different number of days by setting the **Days to Preserve Standard Data Sets Before Deleting** configuration parameter.
2. Reviews the number of data lines still in the stage tables. If the number exceeds a specified maximum value, then data sets may be archived automatically. You specify:
  - o The maximum number of data lines on the **Maximum Data Lines Before Archiving** configuration parameter.
  - o The number of days since a data set was last updated before it can be backed up. Specify this value on the **Days to Preserve Standard Data Sets Before Archiving** configuration parameter.

Therefore, a data set is a candidate for automatic archiving only if:

- The number of data lines in the stage tables exceeds the value specified on the **Maximum Data Lines Before Archiving** parameter.
- The data set hasn't been updated in the number of days specified on the **Days to Preserve Standard Data Sets Before Archiving** parameter.

The archive starts with the oldest data set and continues until the number of data lines in the stage tables drops below the specified maximum.

The same process is used to delete extended retention data sets. The retention period of extended retention data sets is controlled by the Days to Preserve Extended Retention Data Sets.

## How Delete HCM Spreadsheet Data Loader Stage Table Data process deletes spreadsheet data sets?

When you run HCM Spreadsheet Data Loader (HSDL), the system checks whether a schedule exists for the **Delete HCM Spreadsheet Data Loader Stage Table Data** process. If none exists, a new schedule is automatically created to run weekly.

This scheduled process uses the **Days to Retain Data Sets Before Deleting** parameter to purge data sets that haven't been processed within the specified time. The default value of this parameter is 60 days, but you can customize it on the **HCM Spreadsheet Data Loader Parameters** tab of the **Configure HCM Data Loader** page available in the **Data Exchange** work area.

**Note:**

- When the HSDL data set is removed from the HDL staging tables, it can no longer be used for processing. It's only useful for reporting purposes when retained beyond its expiry period.
- If the HSDL purge process fails more than five consecutive times, the **Cancel Failing Delete HCM Data Loader Stage Data Schedules** process will cancel the **Delete HCM Spreadsheet Data Loader Stage Table Data** schedule.

## What happens to the archived data sets?

When a data set is archived, the only values that remain in the stage tables are - statuses of the data sets and business objects, record count, and all associated error messages. All the other related data for the data set is deleted.

After a data set is archived:

- You can't open error management pages for that data set.
- It doesn't appear in search results when you search for data sets to purge.
- You can access the data using a custom query only.

You can archive data sets manually by running the **Delete HCM Data Loader Stage Table Data** process with the **Action** parameter set to **Archive**. You can't create a schedule when the action is **Archive**.

## How automatic maintenance of the stage tables works in conjunction with a user request to delete stage table data?

When you perform the Delete Stage Table Data task, it runs the Delete HCM Data Loader Stage Table Data process. This process maintains the HCM Data Loader stage tables automatically.

This example shows how automatic maintenance of the stage tables works in conjunction with a user request to delete stage table data.

### Maintaining the Stage Tables Automatically

The user performs the **Delete Stage Table Data** task on 9 May 2018 to purge any data sets:

- That weren't updated in the last 5 days
- Whose **Import** and **Load** statuses are both **Success**

Configuration parameters have their default values, as follows:

Configuration Parameter	Value
Days to Retain Data Sets Before Archiving	5

Configuration Parameter	Value
Days to Retain Data Sets Before Deleting	30

The following table shows the status of recent import and load processes when the user performs **Delete Stage Table Data**.

Data Set	Imported Status	Loaded Status	Last Updated
1	Success	Success	9 May 2018
2	Success	Error	7 May 2018
3	Success	Success	1 May 2018
4	Success	Error	28 April 2018
5	Success	Success	8 April 2018
6	Success	Error	8 April 2018

These data sets are either purged or backed up as follows:

1. Data sets 3 and 5 are purged by the user's process, based on the user's parameters.
2. Data set 6 is purged automatically because it hasn't been updated in the last 30 days.
3. Data set 4 is backed up automatically. The backup occurs because the number of data lines in the stage tables still exceeds the configured **Maximum Data Lines Before Archiving** value.

## How do I maintain the stage tables for HCM Data Loader?

You can maintain the stage tables for HCM Data Loader by viewing the stage table status and then either deleting data sets immediately or scheduling a recurring deletion request.

### How to Check the Stage Table Status

Follow these steps to check how full the stage tables are:

1. In the **Data Exchange** work area, click **Delete Stage Table Data**.
2. Check the value on the **Stage Table Capacity Used** section. This percentage helps to know the capacity left on the stage tables. Based on this value, you can decide to delete data sets.

**Tip:** The stage table capacity information is also available on the **Recent File Loads** task.

## Delete a Single Data Set on the Import and Load Data Page

Follow these steps to delete a single data set from the stage tables on the Import and Load Data page:

1. In the **Data Exchange** work area, select **Import and Load Data**.
2. Search for and select the data set that you want to delete.
3. In the Search Results section, click the **Delete** icon for the selected data set.

**Note:** You can only delete protected data sets if you have the `Delete HCM Data Loader Protected Data-Sets` function security privilege.

4. In the Schedule Request dialog box, set the **Delete Source File** parameter to **Yes** to delete the source file from the Oracle WebCenter Content server.

**Note:** You can delete the data from the stage tables without also deleting the source file. If you loaded this data set with the **Delete Source File** parameter set to **Yes**, then the source file has already been deleted. In this case, selecting **Yes** in the Schedule Request dialog box has no effect.

5. Click **Submit**. The ID of the deletion process appears.
6. Click **OK** to close the Confirmation dialog box.

## Delete Multiple Data Sets on the Delete Stage Table Data Page

You can delete one or more data sets immediately from the stage tables. Alternatively, you can schedule a recurring deletion process. Follow these steps:

1. In the **Data Exchange** work area, select **Delete Stage Table Data**.
2. On the Delete Stage Table Data page, set the filter criteria to identify the data sets you that you want to delete. This table describes the main criteria.

Field	Description
Days Since Last Updated	Specify the number of days since the data set was last updated. For example, to identify any data set not updated in the last week, set this field to <b>7</b> . The default value is 30, the maximum value is 60, and the minimum value is 1.
Type	Select any one of the following: <ul style="list-style-type: none"> <li>○ Standard</li> <li>○ Extended</li> </ul> This field is hidden unless you have the additional privilege (Delete HCM Data Loader Extended Retention Data sets) required to delete extended retention data sets.
Created By	Enter the user name of the user who created the data set.

Field	Description
File Type	Select one of: <ul style="list-style-type: none"> <li>○ Compressed DAT file</li> <li>○ Spreadsheet</li> </ul>
Import Status	Select a status. For example, set to <b>Success</b> to list data sets that imported successfully.
Load Status	Select a status. For example, set to <b>Success</b> to list data sets that loaded successfully.
Data Set	Specify a particular data set.
Content ID	Enter the Oracle WebCenter Content ID.
Retention Period	By default, only data sets that have the standard retention period will be deleted. When set to <b>Extended</b> , data sets that are held for an extended period in the staging tables are deleted. This field is hidden unless you have the additional privilege (Delete HCM Data Loader Extended Retention Data Sets) required to delete extended retention data sets.

3. To delete all data sets that match the criteria specified:

- a. Click **Actions > Schedule Deletion**.
- b. On the **Parameters** tab of the **Schedule Request** page, the data set criteria from the **Delete Stage Table Data** page appear. You can't edit these values on the **Schedule Request** page.

**Tip:** To change the data set criteria, edit them on the **Delete Stage Table Data** page, review the results, and click **Actions > Schedule Deletion** again. You must refresh the search results before clicking **Actions > Schedule Deletion**. Otherwise, the parameters aren't updated on the **Parameters** tab of the **Schedule Request** page.

- c. Review and set the **Action** parameter.
- d. Review and set the **Delete Oracle Search Bulk Import Records** parameter.
- e. Review and set the **Initiate Multithreaded Processing** parameter.

**Note:** This parameter ensures that the elapsed time for deleting large volumes of data is reduced by submitting the **Delete HCM Data Loader Stage Table Data** process as a multi-threaded process. Only if the volume of data to be deleted warrants it will the process be multi-threaded.

- f. Source files for the data sets are deleted from the Oracle WebCenter Content server by default. To retain the source files, set the **Delete Source File** parameter to **No**.

4. In the **Schedule** section,
  - a. To submit the delete process immediately, set **Run** to **As soon as possible**.
  - b. To schedule a recurring delete process for all data sets that match the specified criteria:
    - i. Set **Run** to **Using a schedule**
    - ii. Specify the frequency of the deletion process. For example, you could schedule the process to run weekly to delete any data set that hasn't been updated in the last 7 days.
    - iii. Enter the **Start Date** and **End Date**

**Note:** If the start date is tomorrow's date, then the process runs tomorrow and at the specified interval (for example, weekly) until the end date. If the start date is today's date, then the process runs for the first time in one week's time, assuming a weekly frequency.

5. Click **Submit**.

You can review the status of any deletion process by clicking **Actions > Review Processes** on the **Delete Stage Table Data** page. On the **Recent Processes** page:

- To view the delete process parameters such as **Status**, **Submission Time**, **Scheduled Time**, and **Elapsed Time**, action icon for the process and select **Parameters**. If the process created a schedule, then the **Parameters** dialog box also includes the schedule details. Scheduled processes are identified by an icon before the process name.
- You can cancel a schedule for any scheduled process by clicking the action icon for the process and selecting **Cancel Schedule**.
- Recurring deletion requests appear on the Recent Processes page with a parent process in Wait status. The parent process starts a child process for each scheduled deletion request. The ID of the parent process appears in the Scheduling Process ID column.
- For deletions involving 500 or fewer data sets, the process log file includes the request parameters and the details of each data set. If more than 500 data sets are deleted, the log file provides the total number of deleted data sets along with an SQL statement to retrieve the details of the purged data.

To access the log file, click the action icon for the process and select **Log**.

## Delete Spreadsheet Data Sets

On the Delete Stage Table Data page, your search results include spreadsheet data sets if you set **File Type** to **Spreadsheet**. You can delete these data sets from the HCM Data Loader stage tables. But the spreadsheet data sets remain in the HCM Spreadsheet Data Loader stage tables. So, you can continue to edit and upload these spreadsheet data sets after you have deleted them from the HCM Data Loader stage tables. Such a spreadsheet is reloaded from the HCM Spreadsheet Data Loader stage tables when you next save or upload it. To delete spreadsheet data sets from both sets of stage tables, use the **Delete Spreadsheet Stage Table Data** task.

## How do I maintain the stage tables for HCM Spreadsheet Data Loader?

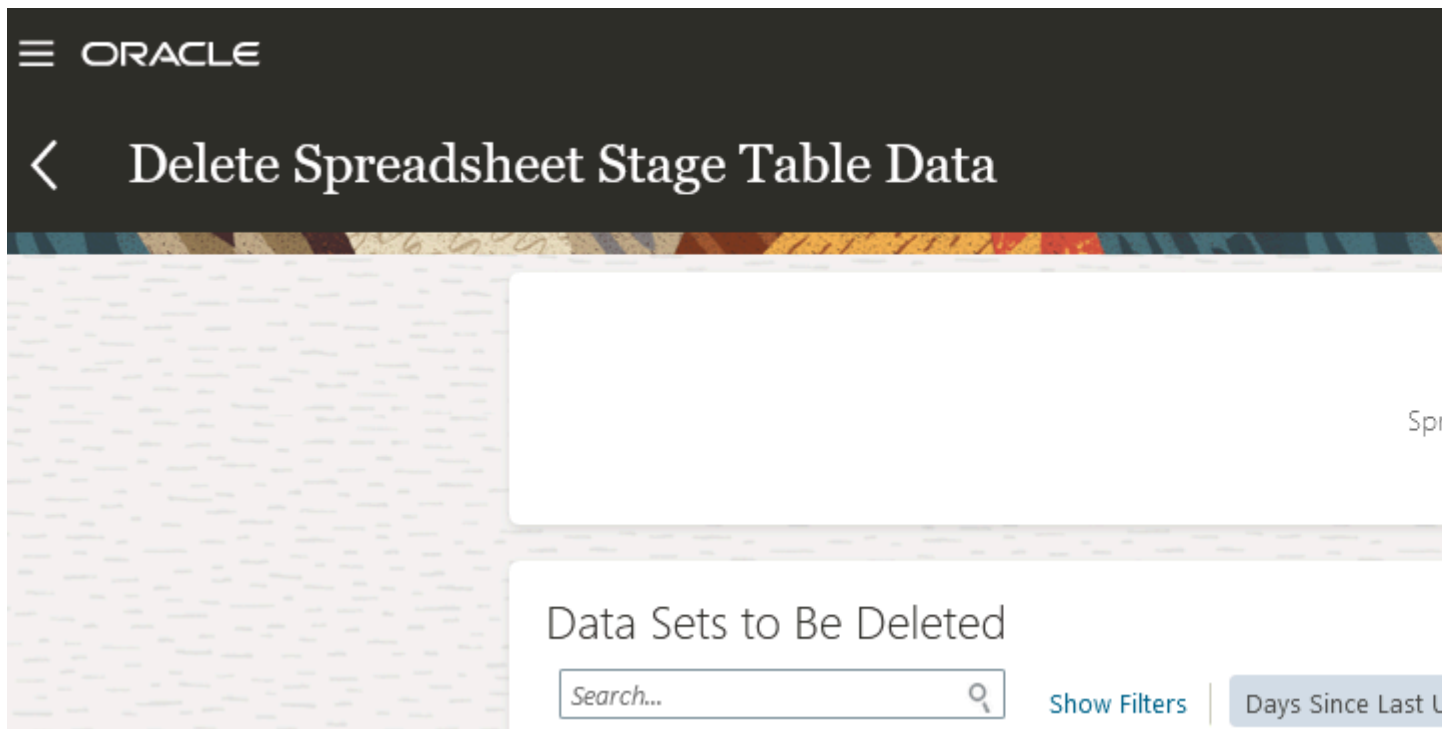
You can either delete data sets immediately or schedule a recurring deletion request.

**Note:** When you delete HCM Spreadsheet Data Loader data sets, they are deleted from all stage tables.

## How to Check the Stage Table Status

Follow these steps to check how full the stage tables are:

1. In the **Data Exchange** work area, select **Delete Spreadsheet Stage Table Data**.
2. Check the value on the **Spreadsheet Stage Table Capacity Used** section.



This percentage helps to know the capacity left on the stage tables. Based on this value, you can decide to delete data sets.

## Delete Data Sets from the HCM Spreadsheet Data Loader Stage Tables

Follow these steps:

1. In the **Data Exchange** work area, select **Delete Spreadsheet Stage Table Data**.

2. On the **Delete Spreadsheet Stage Table Data** page, set the filter criteria to identify the data sets that you want to delete. This table describes the criteria:

Field	Description
Days Since Last Updated	Specify the number of days since the data set was last updated. For example, to identify any data set not updated in the last week, set this field to <b>7</b> . The default value is 30, the maximum value is 60, and the minimum value is 1.
Template	Select an HCM Spreadsheet Data Loader template to identify any data set created in spreadsheets generated from the template.
Data Set	Specify a particular data set. If multiple spreadsheets have been used to load the data set, then they're all listed.
Status	Select one of: <ul style="list-style-type: none"> <li>○ Import completed with errors</li> <li>○ Load completed with errors</li> <li>○ Load successful</li> <li>○ Ready to process</li> </ul>
Last Updated By	Enter the user name of the user who last updated the data set.
Created By	Enter the user name of the user who created the data set.

3. To delete all data sets that match the criteria specified:
  - a. Click **Actions > Schedule Deletion**.
  - b. On the **Parameters** tab of the **Schedule Request** page, the data set criteria from the **Delete Spreadsheet Stage Table Data** page appears. You can't edit these values on the **Schedule Request** page.
 

**Tip:** To change the data set criteria, edit them on the **Delete Spreadsheet Stage Table Data** page, review the results and click **Actions > Schedule Deletion** again. You must refresh the search results before clicking **Actions > Schedule Deletion**. Otherwise, the parameters won't be updated on the **Parameters** tab of the **Schedule Request** page.
  - c. Click **Submit**.
4. To schedule a recurring deletion process for all data sets that match the specified criteria:
  - a. Click **Actions > Schedule Deletion** on the **Delete Spreadsheet Stage Table Data** page.
  - b. Set **Run** to **Using a schedule**.
  - c. Specify the frequency of the deletion process. For example, you could schedule the process to run weekly to delete any data set that hasn't been updated in the last 7 days.
  - d. Enter start and end dates.

**Note:** If the start date is tomorrow's date, then the process runs tomorrow and at the specified frequency (for example, weekly) until the end date. If the start date is today's date, then the process runs for the first time in one week's time, assuming a weekly frequency.

e. Click **Submit**.

You can review the status of any deletion process by clicking **Actions > Review Processes** on the **Delete Spreadsheet Stage Table Data** page. On the **Processes** page:

- You can see the parameters for any process by clicking the contextual action icon next to the process name. If the process was scheduled, then the **Parameters** dialog box also includes the schedule details. Scheduled processes are identified by an icon in the **Schedule** column.
- Recurring deletion requests appear on the **Processes** page with a parent process in **Wait** status. The parent process starts a child process for each scheduled deletion request. The ID of the parent process appears in the **Scheduling Process ID** column.
- For deletions involving 500 or fewer data sets, the process log file includes the request parameters and the details of each data set. If more than 500 data sets are deleted, the log file provides the total number of deleted data sets along with an SQL statement to retrieve the details of the purged data.

To access the log file, click the action icon for the process and select **Log**.

When you delete data sets from the **HCM Spreadsheet Data Loader** stage tables, they're also deleted from the **HCM Data Loader** stage tables.

## What are the recommended HCM Data Loader best practices?

Follow these recommendations to ensure successful creation and maintenance of data using HCM Data Loader:

<p>Source Keys</p>	<p>The source key is the recommended key type to use for all implementations. If you supply a source key when you create your data, then you can also update or simply reference that data using source keys.</p> <p>You can't update some objects if you identify them using only user keys. For example, you can't update most components of the Worker object if you supply only a user key. The attribute that you want to change is often the attribute that's used to identify the record. For example, in the Person Address component, the <b>AddressLine1</b> attribute is used both to identify the address to update and to supply the new value. Therefore, you're recommended to use source keys whenever possible. You can also update source keys. You may want to do this when the source key value is auto-generated.</p>
<p>Data Migration</p>	<p>When creating business objects using HCM Data Loader:</p> <ul style="list-style-type: none"> <li>• Deliver one object type in each .zip file. For example, create one .zip file for jobs, one for grades, one for workers, and so on. To avoid data-reference errors, you must ensure that you load .zip files in the correct order and fix any errors before loading the next file.</li> <li>• Purge the stage table data regularly for data sets that you no longer need, especially if you're loading large volumes of data.</li> </ul>

	<ul style="list-style-type: none"> <li>Understand and follow the rules for creating date-effective data.</li> </ul> <p><b>Note:</b> HCM Data Loader can load very large volumes of data from a single file. However, do consider the practical challenges, such as error management and the length of the transition period, associated with large data loads. Best practice is to split large loads into logical groupings so that you can extract, transform, and load the data more easily. For example, when loading workers you could load ex-employees separately from current, active employees. If this grouping still leaves you with very large data loads, then you could define subgroups of records based on other significant criteria.</p>
Updating Objects	<p>When updating business objects using HCM Data Loader:</p> <ul style="list-style-type: none"> <li>Understand and follow the rules for maintaining date-effective data. In particular, always include the <b>SET PURGE_FUTURE_CHANGES N</b> instruction in the .dat file to ensure that any existing future-dated changes are retained.</li> <li>Supply all business-object files in the same .zip file. HCM Data Loader processes them in the correct order. Referenced data is loaded before the data that references it.</li> <li>Supply any changed attribute values and a unique reference to the record that you're updating. HCM Data Loader validates every attribute name in the <b>METADATA</b> line.</li> <li>Never include both <b>DELETE</b> and <b>MERGE</b> instructions for the same record in the same file. HCM Data Loader doesn't guarantee the order in which the file lines are processed.</li> </ul>

## What are the folders do I use to create your reports within OTBI and BI Publisher?

The following folders are available in the HCM Integrations Real Time subject area. Use these folders to create your reports within OTBI and BI Publisher

### HCM Data Loader

The HCM Data Loader area contains folders relating to HCM Data Loader (HDL) and HCM Spreadsheet Data Loader (HSDL) data sets and processes and HSDL templates.

Folder Name	Description
Data Sets	HCM Data Loader and HCM Spreadsheet Data Loader data set details, such as created by, last update date, and import and load statuses.
Data Set Business Objects	Status information for the business objects processed within HCM Data Loader and HCM Spreadsheet Data Loader data sets.
Data Set Business Object Parameters	Details of the overridden parameters for a business object within a data set.
Data Set Post Processes	Details of the automatically submitted processes for your data set.

Folder Name	Description
Data Set Business Object Post Processes	Details of post processes that are automatically submitted after data for the business object has been successfully loaded.
Facts	Numeric facts for a business object's files, lines, and imported objects.
Imported Objects	Data related to the objects imported for a business object within a data set. It includes the identifying key, reference attribute values with the line, and object messages. This data is only available while the data set exists in the stage tables.
Processes	Information relating to the processes used by a data set.
Process Parameters	Parameter values used when processing a data set.
Spreadsheet Data Sets	HCM Spreadsheet Data Loader data sets details and facts.
Spreadsheet Templates	HCM Spreadsheet Data Loader template information.
Spreadsheet Template Roles	Roles associated with HCM Spreadsheet Data Loader templates.
Spreadsheet Template Parameters	Overridden parameters for HCM Spreadsheet Data Loader templates.

### HCM Data Disposal

The HCM Data Disposal area contains folders relating to the Remove Person Information functionality.

Folder Name	Description
Remove Person Information	Information relating to Remove Person Information templates and processes.
Remove Person Information Facts	Number of processes generated by the Remove Person Information functionality.

## How do I change the first effective start date or last effective end date for some business objects?

For some objects, you can change the first effective start date or last effective end date.

To change an object's first effective start date, you include the **ReplaceFirstEffectiveStartDate** attribute in the .dat file and set it to **Y**. To change an object's last effective end date, you include the **ReplaceLastEffectiveEndDate** attribute in the .dat file and set it to **Y**. You supply the revised dates on the **EffectiveStartDate** or **EffectiveEndDate** attribute, as appropriate. You can make these changes at the same time as other changes. For example, you can change both the start date of an object and its name.

**Note:** The **ReplaceFirstEffectiveStartDate** and **ReplaceLastEffectiveEndDate** parameters don't appear in object template files or on the Attributes tab of the Business Object Details page. However, you can add them to .dat files for date-effective objects that support these changes.

These restrictions apply to changing first effective start date and last effective end date:

- For some objects, the last effective end date must be 4712/12/31. Any attempt to change this date causes an error.
- You can't change the first effective start date of the Worker object in this way. Instead, you must use the **NewStartDate** attribute.
- When you include the **ReplaceLastEffectiveEndDate** attribute in a .dat file, you must specify a date value for the **EffectiveEndDate** attribute. You must not supply a #RETAIN, #ALL, or #NULL token in place of the effective end date.

**Tip:** On the Component Details tab of the Business Object Details page, identify whether you can change a component's first effective start or last effective end date. For example, in the Date Effectivity section on the Component Details tab for the Job object, the **First start date** option is selected. Therefore, you can include the **ReplaceFirstEffectiveStartDate** parameter in the .dat file.

## Changing the First Effective Start Date of a Job Object

This example shows how to include the **ReplaceFirstEffectiveStartDate** attribute in a .dat file.

```
METADATA | Job | JobId | EffectiveStartDate | EffectiveEndDate | ReplaceFirstEffectiveStartDate  
MERGE | Job | 23452 | 1950/01/01 | 4712/12/31 | Y
```

## Changing the Last Effective End Date of a Recurring Element Entry

This example shows how to include the **ReplaceLastEffectiveEndDate** attribute in a .dat file.

```
METADATA | ElementEntry | ElementEntryId | EffectiveStartDate | EffectiveEndDate | ReplaceLastEffectiveEndDate  
MERGE | ElementEntry | 4634 | 2014/01/01 | 2014/04/05 | Y
```

## What are the guidelines for loading descriptive and extensible flexfield data?

Using HCM Data Loader, you can load data for both descriptive flexfields and extensible flexfields. This topic explains concepts that are common to both types of flexfields.

## Configuring Flexfields

Before you generate template files for objects with flexfields, you must configure the flexfields in Oracle HCM Cloud. After the flexfields are configured, the METADATA lines in generated template files include all attributes that are required to load the flexfield data. You also see a Flexfield Attributes tab for relevant object components on the Business Object Details page in the Data Exchange work area.

## Flexfield Codes

When loading flexfield data, you must supply the flexfield code in the METADATA line in this format:

```
FLEX:<flexfield code>
```

For example, to load data for the job descriptive flexfield, the flexfield code is:

```
FLEX:PER_JOBS_DFF
```

Relevant flexfield codes appear in the generated template file and on the Flexfield Attributes tab for the business object.

## Flexfield Attributes

Flexfield attribute names are those that you specify when configuring the flexfield. Both descriptive and extensible flexfields have one or more contexts. When you include a flexfield attribute name on the METADATA line for an object, you must also identify the context. You provide a hint value in the following format to identify both the flexfield and the context to which the attribute belongs:

```
<flexfield attribute name>(<flexfield code>=<context code>)
```

For example, for the `_MAIN_CONTRACT` attribute of the `PER_CONTRACT_LEG_DDF` descriptive flexfield with the `CN` context code, the entry on the METADATA line is:

```
_MAIN_CONTRACT(PER_CONTRACT_LEG_DDF=CN)
```

## Business Objects with Multiple Descriptive Flexfields

Some business object components support multiple descriptive flexfields. You can include all descriptive flexfield attributes for every flexfield and configured context on a single METADATA line. This approach is possible because you supply both the flexfield code and the context for each attribute.

## Lookup and Value Set Validated Flexfield Values

For flexfield values that are validated by lookups, you can supply either the lookup code or the lookup value. To supply the value, you must append the suffix `_Display` to the attribute name in the METADATA line.

For example, for a configured gender flexfield attribute, you can supply either the lookup code (M or F) or the value (Male or Female). To supply the value, use the attribute name `gender_Display`, as shown in this example:

```
METADATA|Job|FLEX:PER_JOBS_DFF|gender_Display(PER_JOBS_DFF=JOBCONTEXT1)|...  
MERGE|Job|JOBCONTEXT1|Male|...  
MERGE|Job|JOBCONTEXT1|Female|...
```

The same is true for flexfield segments validated by table-validated value sets. You can supply either the ID or value. To supply the value, you must append the suffix `_Display` to the attribute name in the METADATA line.

**Note:** You're recommended to use the lookup code or value set ID, as the lookup meaning and value set value can be translated. The language of the value provided to the `_Display` attribute must match the language of the user who's uploading the data. For HCM Data Loader that's the elevated `FUSION_APPS_HCM_ESS_LOADER_APPID` user, whose language may differ from language of the session user.

## Secured Value Sets

A value set is a group of values that validate the content of a flexfield segment. If the value set is security enabled, then it's available to specific users or processes only. You must ensure that the predefined `FUSION_APPS_HCM_ESS_LOADER_APPID` user has access to any security-enabled value sets that your flexfields reference. Otherwise, the load fails. For more information about enabling access to secured value sets, see the Securing HCM guide.

## Deleting Flexfield Segments

Let's say, for GDPR or other compliance purposes, you need to remove values from a 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.

Here's a tutorial on loading flexfield segments using HDL: [Loading Flexfield Segments with HCM Data Loader \(HDL\)](#)

Here's a tutorial to add preconfigured and custom flexfield segments to your HCM Spreadsheet Data Loader templates [Including Flexfield Segments in HCM Spreadsheet Data Loader Templates](#)

## How do I include descriptive flexfield data using METADATA and MERGE lines in a .dat file?

This topic describes how to construct METADATA and MERGE lines in a .dat file to supply descriptive flexfield data.

### Constructing the METADATA Line

Descriptive flexfields extend a business object. You can supply the attributes of the descriptive flexfield on the same MERGE line as the core attributes of the business-object component. On the METADATA line, you must include both the flexfield code and the descriptive flexfield attributes for which you want to load data. In a generated business-object template, the METADATA line automatically includes the flexfield code and attributes for configured descriptive flexfields.

This example shows the METADATA line for the Job business object with entries for the PER\_JOBS\_DFF descriptive flexfield:

```
METADATA|Job|FLEX:PER_JOBS_DFF|JobCode|ActiveStatus|FullPartTime|MedicalCheckupRequired|RegularTemporary|
EffectiveStartDate|EffectiveEndDate|Name|SetCode|JobFamilyName|_JOB_LEVEL(PER_JOBS_DFF=US)
```

The descriptive flexfield attributes can appear anywhere on the line. You don't have to append them at the end.

### Constructing the MERGE Lines

For each descriptive flexfield for a business-object component, a single MERGE record can have only one context. You specify the context value on the MERGE line against the flexfield code for the descriptive flexfield in the METADATA line.

For example:

```
METADATA|Job|FLEX:PER_JOBS_DFF|JobCode|ActiveStatus|FullPartTime|MedicalCheckupRequired|RegularTemporary|
EffectiveStartDate|EffectiveEndDate|Name|SetCode|JobFamilyName|_JOB_LEVEL(PER_JOBS_DFF=US)
MERGE|Job|US|ACC|A|F|N|R|2000/01/01|4712/12/31|Accountant|COMMON|Finance|1
```

On the MERGE line, the context value (US) appears in the same position as the flexfield code (FLEX:PER\_JOBS\_DFF) on the METADATA line.

## Supplying MERGE Lines for Multiple Flexfield Codes and Contexts

When a single business object component supports multiple descriptive flexfields, you can load all of the flexfield data at once. The following example shows METADATA and MERGE lines for the Contract component of the Worker object. Each MERGE line is for a single context.

```
METADATA|Contract|AssignmentId|ContractId|EffectiveStartDate|EffectiveEndDate|
FLEX:PER_CONTRACT_DF|FLEX:PER_CONTRACT_LEG_DDF|_CONTRACT_GLB(PER_CONTRACT_DF=Global Data
Elements) |_Currency(PER_CONTRACT_DF=CONTRACT_DF) |_MAIN_CONTRACT(PER_CONTRACT_LEG_DDF=CH) |
_CONST_PROB_DATE(PER_CONTRACT_LEG_DDF=CN) |_NDA(PER_CONTRACT_LEG_DDF=CN) |
_COMPETITION_CLAUSE(PER_CONTRACT_LEG_DDF=CN) |_NOTICE_DURATION_UNIT(PER_CONTRACT_LEG_DDF=CN)
MERGE|Contract|E8732|39987|2013/12/14|2014/03/04|CONTRACT_DF|CN|Contract Glb value|USD|Contract Data||
MERGE|Contract|E8732|39987|2014/03/05|4712/12/31|CONTRACT_DF|CH|Contract Glb value|USD||31/03/2015|NDA
Value|Competition Clause Value
```

## Updating Global Segments

You can configure both global and context-specific segments for a flexfield. When updating global segments in records that have context-specific segments, set the context value to the existing context value, not Global Data Elements. If you set the context value to Global Data Elements, then existing context-specific values are set to null because the context has changed. For example, consider these two segments of a flexfield record:

Segment	Global	Context
Segment 1	Yes	Not applicable
Segment 2	No	CN

If you update segment 1, which is a global segment, then you must set the context to CN, not Global Data Elements. Otherwise, the existing contents of segment 2 are set to null.

## Example of Loading Data for Descriptive Flexfields

This section explains how to define the METADATA line for the Contract component of the Worker object with multiple configured descriptive flexfields. The METADATA line appears automatically in the generated worker template.

The contract component supports these descriptive flexfields:

- PER\_CONTRACT\_DF
- PER\_CONTRACT\_LEG\_DDF

## Constructing the METADATA Line for the Worker Contract Component

The PER\_CONTRACT\_DF descriptive flexfield is configured in Oracle HCM Cloud with the context and attribute values shown in this table. The **Hint** column shows the resulting hint for each attribute.

Context	Attribute	Hint
Global	CONTRACT_GLB	(PER_CONTRACT_DF=Global Data Elements)

Context	Attribute	Hint
CONTRACT_DF	Currency	(PER_CONTRACT_DF=CONTRACT_DF)

The PER\_CONTRACT\_LEG\_DDF descriptive flexfield is configured in Oracle HCM Cloud with the context and attribute values shown in this table. The **Hint** column shows the resulting hint for each attribute.

Context	Attribute	Hint
CH	_MAIN_CONTRACT	(PER_CONTRACT_LEG_DDF=CH)
CN	_CONST_PROB_DATE	(PER_CONTRACT_LEG_DDF=CN)
CN	_NDA	(PER_CONTRACT_LEG_DDF=CN)
CN	_COMPETITION_CLAUSE	(PER_CONTRACT_LEG_DDF=CN)
CN	_NOTICE_DURATION_UNIT	(PER_CONTRACT_LEG_DDF=CN)

In the generated Worker.dat template file, the Contract METADATA line includes these attribute definitions for the flexfield attributes:

```
METADATA|Contract|...|_CONTRACT_GLB(PER_CONTRACT_DF=Global Data Elements) |
_Currency(PER_CONTRACT_DF=CONTRACT_DF) | _MAIN_CONTRACT(PER_CONTRACT_LEG_DDF=CH) |
_CONST_PROB_DATE(PER_CONTRACT_LEG_DDF=CN) | _NDA(PER_CONTRACT_LEG_DDF=CN) |
_COMPETITION_CLAUSE(PER_CONTRACT_LEG_DDF=CN) | _NOTICE_DURATION_UNIT(PER_CONTRACT_LEG_DDF=CN)
```

As each attribute has a hint that identifies both the descriptive flexfield and the context, you can define attributes for multiple flexfields on the same METADATA line.

## How do I include extensible flexfield data using METADATA and MERGE lines in a .dat file?

This topic describes how to construct METADATA and MERGE lines in a data file to supply extensible flexfield data.

### Category Code

Extensible flexfields have both a flexfield code and a category code. The category code is provided automatically in generated template files and has the attribute name EFF\_CATEGORY\_CODE.

### Constructing the METADATA Line

Unlike descriptive flexfields, extensible flexfields aren't an extension of a business-object component. Rather, they're separate components in the business object hierarchy. To supply data for an extensible flexfield, you simply include the

METADATA line for the extensible flexfield component in the data file. Remove the names of any attributes for which you aren't supplying data.

This example shows a METADATA line for the JobLegislative extensible flexfield.

```
METADATA|JobLegislative|EFF_CATEGORY_CODE|FLEX:PER_JOBS_LEG_EFF|JobId(SourceSystemId)|JobCode|
SetCode|EffectiveStartDate|EffectiveEndDate|SourceSystemOwner|SourceSystemId|LegislationCode|
_EEO1_CATEGORY(PER_JOBS_LEG_EFF=HRX_US_JOBS)|_FLSA_STATUS(PER_JOBS_LEG_EFF=HRX_US_JOBS)|
_INSEE_PCS_CODE(PER_JOBS_LEG_EFF=FR)|_ECAP_JOB(PER_JOBS_LEG_EFF=FR)|
_LINE_OF_PROGRESSION(PER_JOBS_LEG_EFF=HRX_US_JOBS)
```

## Constructing the MERGE Lines

Each MERGE line for an extensible flexfield record can have only one context. In the METADATA line, you specify the context value against the flexfield code for the extensible flexfield.

This example shows a METADATA line and two MERGE lines for the JobLegislative extensible flexfield:

```
METADATA|JobLegislative|EFF_CATEGORY_CODE|FLEX:PER_JOBS_LEG_EFF|JobId(SourceSystemId)|JobCode|
SetCode|EffectiveStartDate|EffectiveEndDate|SourceSystemOwner|SourceSystemId|LegislationCode|
_EEO1_CATEGORY(PER_JOBS_LEG_EFF=HRX_US_JOBS)|_FLSA_STATUS(PER_JOBS_LEG_EFF=HRX_US_JOBS)|
_INSEE_PCS_CODE(PER_JOBS_LEG_EFF=FR)|_ECAP_JOB(PER_JOBS_LEG_EFF=FR)|
_LINE_OF_PROGRESSION(PER_JOBS_LEG_EFF=HRX_US_JOBS)
MERGE|JobLegislative|JOB_LEG|HRX_US_JOBS|OCT18EFF1|OCT18EFF1|COMMON|1990/01/01|4712/12
MERGE|JobLegislative|JOB_LEG|FR|OCT18EFF1|OCT18EFF1|COMMON|1990/01/01|4712/12/31|VISION|OCT18EFF1_LEG2|FR|||
387b|N|
```

**Note:** Unlike other components of the business-object hierarchy, extensible flexfields can't be supplied in isolation. They must be accompanied by a parent record.

## Multirow Extensible Flexfield Contexts

You can configure an extensible flexfield context with multiple rows. In this case, you configure one or more of the flexfield segments as the key that uniquely identifies a single row of the multirow context. HCM Data Loader treats these flexfield segments as user-key attributes. Therefore, you can uniquely identify a single flexfield record using user keys.

You must supply a value for the user-key flexfield-segment attributes when creating an extensible flexfield record. When updating a multirow extensible-flexfield record, you must supply either the source key or the complete user key, including the flexfield segments defined as the unique key. You can identify which flexfield segments form the unique key by reviewing the flexfield-segment configuration on the Manage Extensible Flexfields page. All flexfield segments for which **Unique Key** is selected are user-key attributes.

## Example of Loading Data for Extensible Flexfields

This example shows how the METADATA line is constructed for a Job business object component with the configured JobLegislative extensible flexfield. The METADATA line appears automatically in the generated job template.

## Constructing the METADATA Line for the JobLegislative Extensible Flexfield

The PER\_JOBS\_LEG\_EFF extensible flexfield is configured in Oracle HCM Cloud with the context and attribute values shown in this table. The **Hint** column shows the resulting hint for each attribute.

Context	Attribute	Hint
CA	_EEOG	(PER_JOBS_LEG_EFF=CA)
CA	_NOC_CODE	(PER_JOBS_LEG_EFF=CA)
CH	_POSITION_TYPE	(PER_JOBS_LEG_EFF=CH)
FR	_ECAP_JOB	(PER_JOBS_LEG_EFF=FR)
FR	_INSEE_PCS_EXT_CODE	(PER_JOBS_LEG_EFF=FR)

In the generated Job.dat template file, the JobLegislative METADATA line includes these attribute definitions for the flexfield attributes:

```
METADATA|JobLegislative|...|EFF_CATEGORY_CODE|FLEX:PER_JOBS_LEG_EFF|_EEOG(PER_JOBS_LEG_DFF=CA)|
_NOC_CODE(PER_JOBS_LEG_DFF=CA)|_POSITION_TYPE(PER_JOBS_LEG_DFF=CH)|_ECAP_JOB(PER_JOBS_LEG_DFF=FR)|
_INSEE_PCS_EXT_CODE(PER_JOBS_LEG_DFF=FR)
```

## How do I set up encryption for file transfer?

You use encryption keys to encrypt files for secure transfer between Oracle HCM Cloud and your own servers through the Oracle WebCenter Content server. This PGP-based encryption support is available for secure file transfer using HCM Data Loader and HCM Extracts.

The process for inbound files (into Oracle HCM Cloud) is:

1. You encrypt files using the Oracle HCM Cloud public key.
2. The data-loading process decrypts files using the Oracle HCM Cloud private key.

The process for outbound files (generated by Oracle HCM Cloud) is:

1. Oracle HCM Cloud encrypts files using your public key.
2. You decrypt files using your private key.

Therefore, before you can encrypt or decrypt files, you must:

1. Import your public key to Oracle HCM Cloud.
2. Generate the PGP encryption key pair and download the Oracle HCM Cloud public key.

You can also sign the files, as follows:

- Outbound files are signed using the HCM Cloud private key. You verify these files using the HCM Cloud public key.
- Inbound files are signed using your private key. The data-loading process verifies inbound files using your public key.

## What are the keys used for encryption, decryption, signing, and verification in each supported encryption mode?

This table provides you information on the keys that are used for encryption, decryption, signing, and verification in each supported encryption mode:

Encryption Mode	Encryption Key	Decryption Key	Signing Key	Verification Key
Outbound PGP Signed	customer-key_pub	customer-key_priv	fusion-key_priv	fusion-key_pub
Outbound PGP Unsigned	customer-key_pub	customer-key_priv	N/A	N/A
Inbound PGP Signed	fusion-key_pub	fusion-key_priv	customer-key_priv	customer-key_pub
Inbound PGP Unsigned	fusion-key_pub	fusion-key_priv	N/A	N/A

## How do I import a public key for encrypting outbound files?

You use your public key (customer-key\_pub) for encrypting outbound files. You can decrypt the files using your private key (customer-key\_priv). If you also want outbound files to be signed, then use the HCM Cloud private key (fusion-key\_priv) for signing.

You can verify signed outbound files using the HCM Cloud public key (fusion-key\_pub).

To import the customer public key:

1. Sign in to Oracle HCM Cloud with the IT Security Manager job role or privileges.
2. Select **Navigator > Tools > Security Console** to open the Security Console.
3. Click the Certificates tab to open the Certificates page.
4. Click **Import** to open the Import page.
5. Set **Certificate Type** to **PGP**.
6. In the **Alias** field, enter a name to uniquely identify your key.
7. Click **Browse** to identify the location of the customer public key.
8. Click **Import and Close** to import the public key into the Oracle HCM Cloud keystore.

Your public key now appears on the Security Console Certificates page.

## How do I generate a PGP encryption key pair?

You generate the PGP key pair on the Security Console. You download the public key to encrypt files that are inbound into HCM Cloud (for example, input data files for HCM Data Loader). To sign these inbound files, you can use your private key (customer-key\_priv), which is verified using your public key (customer-key\_pub) in Oracle HCM Cloud.

You must have imported the customer public key.

To generate the PGP Encryption Key Pair:

1. Sign in to Oracle HCM Cloud with the IT Security Manager job role or privileges.
2. Select **Navigator > Tools > Security Console** to open the Security Console.
3. Click the Certificates tab to open the Certificates page.
4. Click **Generate** to open the **Generate** dialog box.
5. In the **Generate** dialog box, set **Certificate Type** to **PGP**.
6. In the **Alias** field, enter **fusion-key**.

**Note:** You must enter **fusion-key** in this field. Otherwise, the encryption APIs can't use this key to decrypt all encrypted inbound files.

7. In the **Passphrase** field, enter a passphrase for the private key. This passphrase is needed when you edit, delete, or download the private key.

**Note:** If you forget the passphrase, then you may have to raise a service request for help to delete the private key. Once the old key is deleted, you can generate a new key using the process described here.

8. In the **Key Algorithm** field, select **RSA**.
9. In the **Key Length** field, select either **1024** or **2048**.
10. Click **Save and Close**. The fusion-key pair is generated and ready for download. You can see the fusion-key pair on the Certificates page of the Security Console.
11. In the **Status** actions for the fusion-key pair on the Certificates page, select **Export > Public key**. Save the HCM Cloud public key (**fusion-key\_pub.asc**) to your desktop. Use the downloaded key to encrypt files that are inbound to Oracle HCM Cloud.

## Why must I use the alias fusion-key when generating the PGP key pair?

The encryption and decryption APIs in Oracle HCM Cloud recognize only the alias fusion-key. If you use a different alias, inbound files can't be decrypted correctly.

## What's the role of the passphrase when creating a key pair?

The pass phrase protects the private key. You'll need it if you want to edit, delete, or download the private key later. If it's lost, the private key may have to be deleted and a new key pair generated.

## How do I encrypt and upload files automatically?

You can use REST APIs or command-line tools such as GPG to encrypt files with a supported algorithm and then upload them automatically to Oracle HCM Cloud through the Oracle WebCenter Content server.

The process involves encrypting the file using the HCM Cloud public key (for inbound), optionally signing it, and then transferring it to the content server location configured for HCM file loading.

## What are encryption algorithms are supported by Oracle HCM Cloud?

Oracle HCM Cloud supports the following encryption algorithms -

- Cipher: AES-128, Blowfish, CAST5, 3DES, AES-256
  - | **Note:** These cipher algorithms aren't supported: Twofish, IDEA, AES-192
- Compression: bzip2, zlib, .zip, uncompressed
- Hash: SHA-1, SHA-256, SHA-224, SHA-512, MD5, SHA-384, RIPEMD-160

## How do I encrypt files in Microsoft Windows and Linux environments?

- Gpg4win, the official GnuPG distribution for Microsoft Windows, provides both a command-line interface and a graphical user interface for encryption, decryption, signing, and verification. For encryption, use the command-line interface. You can find Gpg4win here: <https://www.gpg4win.org/about.html>.
- You can download GnuPG for Linux from various sources, depending on the Linux distribution that you're using. Commonly used GnuPG versions can be found here: <https://www.gnupg.org/index.html>.

After installing the Gpg4win or GnuPG tool, follow these steps to encrypt or encrypt and sign a file:

1. Import the HCM Cloud public key (downloaded from the Security Console) using this command at the command prompt:

```
gpg --import <fusion public key file>
```

2.
  - o To encrypt a file without signing, use this command:

```
gpg --cipher-algo <one_of_the_supported_algorithms> -r fusion-key --encrypt <inbound_file_name>
```

- o To both encrypt and sign a file, use this command:

```
gpg --cipher-algo <one_of_the_supported_algorithms> -r fusion-key -se <inbound_file_name>
```

**Note:** When signing files, ensure that your private key is imported into the keystore that's used for signing.

## How do I load encrypted files to Oracle HCM Cloud from the Oracle WebCenter Content server?

After you upload the encrypted files to the Oracle WebCenter Content server, submit the HCM Data Loader Import and Load Data.

You need to pass the encryption type parameter with the encrypted file so that HDL can process the file appropriately. This parameter has

This parameter has the following values, which are defined in the `ORA_HRC_FILE_ENCRYPT_TYPE` lookup type:

- NONE
- PGPSIGNED
- PGPUNSIGNED

