Oracle® Workforce Scheduling

Integration Functional Guide

Release 5.0.3 for Windows

Part No. E12664-01

July 2008

ORACLE®

Oracle® Workforce Scheduling Integration Functional Guide, Release 5.0.3 for Windows

Part No. E12664-01

Contents

Introduction

This guide presents a functional view about the data that can be exchanged through messages (imported and exported) using Oracle Workforce Scheduling (OWS.) It is designed to help designers or developers of an external application map the data that is exchanged between the two applications. However, this guide does not explain how messages are formed, the message syntax, or how messages are produced and exchanged.

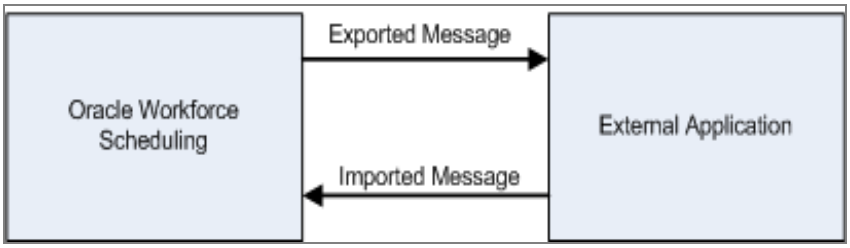

Figure 1: Communication Flow between OWS and External Applications

In general, it is easier to design and develop interfaces, once the designer or developer understands the data that needs to be mapped. Designers or developers must recognize the consistencies, or identify the differences of concepts between their own local source application and OWS. For example, what is referred to as an "employee" in OWS is perhaps not the same concept and corresponds to the "work period of an employee" in the external application. A consultant defines the customer's basic functional configuration of the application in an XML rich document called the CCD (Customer's Configuration Document).

To aid designers and developers in understanding the terminology used in OWS, the chapter "Overview of Key Concepts of Oracle Workforce Scheduling" introduces several terms and their definitions. This chapter helps familiarize the designer and developer with the key concepts used in OWS, and also serves as a glossary.

The "Data Exchanges" describes all precise data. This helps the designers or developers of the external application map the terminology or concepts, based on records and fields, with that of Oracle Workforce Scheduling, before designing the interface. Data is organized as a sequence of records, with each record having a list of fields. These concepts are well known and hence not explained again.

The "Charts" chapter contains numerous examples on the usage of charts in OWS.

The "Exports of Oracle Workforce Scheduling" chapter explains the types of messages that the developer can export from OWS to an external application.

This guide explains parts of the interface that are generic and not related to any specific customer configuration, such as employees, employees' files, business units,

data, hierarchies, absences, and schedules. Therefore, as a developer you will be able to understand most parts of this guide even if you do not have a precise knowledge of the functional configuration of the application.

Overview of Key Concepts of Oracle Workforce Scheduling

This chapter introduces several terms and their definitions to help you understand the key concepts of Oracle Workforce Scheduling (OWS) better.

Customer Configuration

The customer configuration defines various parameters, such as contracts that define employee breaks, workload-computing formulas from business drivers, and the fields selected in interface records. A consultant records these parameters in an XML rich document called the Customer's Configuration Document (CCD).

Business Unit Hierarchies and Classes

Each business unit has a class, which is constant. The class describes the unit, for example, a *Region*, *Store*, or *Department*. Each unit is identified by a unique universal code (a store and a department cannot have a same value for its code).

Business units (usually called organization units in OWS) are represented in two hierarchies:

- Business
- Team

OWS represents an organization's business and management relationships through a business and team hierarchy. Employees report to a manager (team hierarchy) and they work for a business (business hierarchy).
Hierarchical links of one unit to both parents (*business* and *team*) have effective dates and you can import these effective dates. You can represent a unit either as a business or a team, but never as both in OWS.

Employees and Files of Employees

An employee number identifies an employee. OWS supports employee identification by means of a Social Security Number (SSN) and a badge number. The employee number is the actual identifier, which cannot be changed. SSN and badge numbers

can be updated. However, only a record of the last current value is maintained and not the history of all the updates.

OWS records some of the identification data that is directly linked to an employee. The (employee) *PersonIdentification* record contains:

- The employee number called *HRID* as its key.
- Fields such as *FirstName* (mandatory), *LastName* (mandatory), *Title* (Mr, Ms), *SSN*, *Badge number*, *BirthDate*.

Note: OWS maintains a history of all employee records, except the *PersonIdentification* record. All these records have *Effective date* as the last key field.

A person can be rehired several times by the same company. Each work period is represented in OWS by a file. The identification of this file can be either relative to the employee number (serial number or date) or absolute (such as, company number or requisition number).

In some companies, an employee can have only one valid work period at any given point of time, and consequently have only a single job. In this case, the file identification is always set to 1. OWS does not use this information while generating or managing schedules.

In other companies, an employee can have more than one job at a given point of time. For example, an employee may work for a store in the mornings from Monday to Wednesday. The same employee may work in another store during the weekends (having signed two contracts). In this case, you must configure the OWS application to recognize files related to the first job or the second while generating schedules. The file identification in this case is mandatory and must be unique.

OWS maintains file numbers of a given employee, even if the employee only holds a single job at a given period. The application records and manages these file numbers for data import and export.

Demand: Activities, Tasks, and Drivers

A key objective of the OWS application is to generate optimized schedules for employees. A weekly schedule of an employee contains a list of shifts, with each shift mainly defined by the day, start and end times, and the activity performed during this time (or absence).

Activities are specific to each employee, such as cashier, cleaning, and truck delivery.

For the application to generate schedules, you must provide the detailed workload of a store. The following is an example of how to calculate the workload for a store: How many *man*hours* of *cashier* activity must be covered every 15-minutes, in a week?

The cashier activity is composed of several smaller tasks, such as checking items and receiving cash or card payments. The effort required to complete these tasks

depends on drivers (the number of items, the number of card payments) and associated labor standards.

Note: Shifts are defined at the activity level, but the workload is computed by task.

Each task has a corresponding time window. The time window contains the start and an end time for the task, for each day of the week. The workload is 0 outside the task time window. Typically, time frame values in time windows are cyclic, linked to an organization, and do not change every day.

You can describe tasks with a time window and a workload value. For example, a management task has a known and fixed workload of 2 hours and you can distribute it anywhere in the time window. Therefore, a time window can also hold a value.

To generate schedules, OWS must compute costs by using the following information for each employee:

- Actual number of hours spent on the task each day
- Cost by day

Forecasting the Workload

Forecasting data for every 15-minute interval in a day produces too much detail and is not useful. The demand computation uses a 15-minute profile, a chart that defines how a daily global effort is distributed for every quarter of the day.

A daily profile of a week has a global weekly value (for example, the *number of items by week*). To get a value for each day, apply a 7-value profile that distributes the value across each day of the week.

OWS maintains a record of all driver values (such as, number of items) for previous weeks and transforms a set of historical actual values into a forecast value. The application does this by doing a trend analysis on the historical values.

OWS also imports forecast values generated by external applications.

Analytics for Import and Export

Oracle Workforce Scheduling uses computed schedules to calculate KPIs on a daily or weekly basis for an organization unit. To import data required for demand computing, OWS uses weekly, daily, and cyclic charts (profiles). Daily charts have two types of information:

- Per day range with up to 96 values corresponding to each quarter of the day.
- Per week range with up to 7 values (corresponding to each day of the week).

OWS uses time windows to import the time limits for tasks. To import absences, OWS uses schedules. A schedule is a list of shifts, which includes both activity and absences for an employee.

To export schedules, OWS uses schedules with multiple shifts by day, one shift by activity or by absence. To export organization KPIs, the OWS application uses charts.

Job Management

The OWS application includes job management and scheduling modules that the functional administrator can use.

- Using the Job Management module, administrators can search, view, and edit asynchronous jobs that have been launched for the store or its departments. The administrator can view job details and edit these details, change the scheduled date of a pending job, cancel a pending or running job. Detailed logs supply further information about jobs. Predefined queries as well as advanced search criteria filter and streamline searches for jobs.
- Using the Job Scheduler module, administrators can schedule the ExportSchedule and ExportKPIs procedures and the launch batch processes, including: Forecast, Demand, Check, FireButton, and EarnedHours procedures.

For job scheduling, the interface message contains the job lists and parameters associated with each job.

List of Job Fragment Fields for Job Scheduling

Field Name	Type	Description
PartyID IDType	String	The Level at which the export is done
StartDate	date	The start scope date of the export
EndDate	date	The end scope date of the export
Day	int	The day of export : -1 for Now option in all frequencies, 1 to 31 for Monthly frequency 0 to 6 for weekly frequency (0 is Sunday) 0 for daily frequency
Time	time	Time of the export
BusinessType	String	The BU Level
TeamType	String	Team Level
JobFrequency	Enumeration (Daily, Weekly or Monthly)	The frequency
Unit	Enumeration (Day, Week or Month)	Unit of offset (optional)
BeforeAfter	Enumeration (Before or	Offset before or after execution date (optional)

	After)	
Number	int	Offset (optional)
JobParameter Name	String	Name of the parameter associated with one of the three jobs to launch
JobParameter Val	String	The value of this parameter

User Roles

OWS includes role-based security. You can create roles in addition to those delivered with the application and customize them. A role is based on a profile which is linked to a site map, an organized set of modules and pages. The user then links the role to a login in the OWS application.

The predefined profiles that you can link to a role include: Admin, OWS Admin, Corporate Application Admin, Store Manager, Department Manager, District Manager, Corporate Profile, and Debug. The OWS application also populates these default profiles as roles on the Logins pages. You can create, update, and delete logins.

For each role, you define the site and tree of pages for the site, and then define the user roles, associate the role to a profile, specify the readers (the type of jobs that generate notifications on the to-do list), and specify the role's access options (display page, read or write privileges, hide/display steps).

When importing the logins, the message consists of the user login, its password, validity dates, and the user role associated to it.

The following is the list of modules assigned to each profile.

Process	**Admin**	**OWS Admin**	**Corporate Application Admin**	**Store Manager**	**Corporate Profile**	**Debug**
Weekly Process			X	X		X
Employee Maintenance			X	X		X
Dashboard			X	X		X
Utilities		X	X	X		X
User Management	X					X
Organization management	X					X
Contract management		X	X			X
Export		X	X			X
Store Not Open		X	X			X

Event Management		X	X			X
Fire Button		X	X			X
Corporate Profile					X	X

Charts of Business Units and Employees' Files

There are several units for a chart (data table):

Unit	Description
Weekly chart	A value for each week (all given dates are first days of weeks). If you do not provide any value for a week, the application considers it a null value.
Daily chart	A value for each day. If you do not provide any value for a day, the application considers it a null value.
Range chart	You can provide a value for a day only when its value is different from the day before. If you do not provide values, the value of the day for which it was last given is taken.
15 minutes daily chart	For each day, you can provide an array of 96 values (one for every quarter in the day or more in case of midnight crossing).
15 minutes / day cyclic chart	An array of 96 (or more) values for every 15 minutes to be cyclically repeated for all similar days of all weeks, for a given period of time. For example, you can use such a chart to describe for every 15 minutes the number of customers at the cash counter on Mondays during peak weeks. You can title the chart as follows: *CashProfileMondayPeak.*
Day/Week cyclic chart	One array of 7 values for every day of a week to be cyclically repeated for all the weeks within a specific time period.

As mentioned earlier, you can use charts (data tables) to record daily or weekly data as KPIs, actuals, accruals, and drivers.

The following are some examples of how you can use charts to import information for a business unit:

- Daily number of customers by store.
- Weekly sales by cash for a department.
- Weekly type of week sales indicator (1:low, 2:regular, 3:high) for a store.

The following are some examples of how you can use charts to import information for an employee's file:

- The actual number of hours worked daily.
- The actual daily cost.

You can configure a list of charts by declaring the list of drivers that define the workload in the CCD.

Charts have a variety of uses, relating to business and employee activity. You can import and export charts by interfaces. To import a chart, you must provide:

- The chart code: This code is defined in the CCD and corresponds to each customer activity. For example, *Activity Labor Standard* is a type of chart. The chart gives the number of employees required to process a given amount of work. If a store has 40 activities such as, shipping, cashier, and management you must import 40 charts of the category <*Activity*> *Labor Standard*. You can declare each of the 40 charts in the CCD and identify each chart for example, by the chart type and the activity name (*ActivityLaborStandardCashier*).
- The target of the chart: A chart can have either an organization unit or an employee as a target.
- The time period (first and last dates): The existing chart values outside the period are left unchanged. You can replace those inside the period with the given chart values.
- Values:
  - For all days in the time period, of a daily chart (an array of 96 values for 15 minutes charts).
  - For all weeks (first day) in the time period, of a weekly chart.
  - For all days on which the value changes in the time period, for a range chart.

Note: You can also import cyclic charts (Day/Week or 15 minutes/Day) by providing only the values that have to be repeated.

Records

OWS manages records of various types. All records, but the employee identification, are valid for a time period. You can specify this period at the time of import or creation. These types of records are described as dated records.

- Business units: OWS manages one dated record by business unit class, such as *Store* and *Department*. You can manage organizational hierarchies by such records. Time Window and Cycle are two records that you can attach to business units.

- Employees: OWS manages only one record that is not dated, which is the *PersonIdentification* record. The application also manages only three historical records related to employee namely, *Address, Hiring,* and *Contact*.
- About employees' files: OWS manages dated records for *employees'* files namely, *Contract, Assignment, Skill, Availability,* and *Schedule.*

You cannot configure records or their fields as they are fixed. However, you can add a few customized fields to some records and declare them in the CCD. (These customized fields are not used in computing but display information.)

You can import and export records. To import a dated record, you need to provide:

- The record code.
- The target of the record. A record can have either a business unit or an employee file as a target.
- The time period of the record (first and last date). The existing record values outside the period are left unchanged. You can replace those inside the period with the supplied record values.
- The complete history of dated records in the time period. For each day in the time period for which at least one field value differs from the previously given one, you must provide:
  - The effective date.
  - The value of fields that are not null (blank).

Example of a Dated History

The following is an example of a dated history. The *key* is the store number and the effective date.

Record History: Variable Name = "Store"			
Store	Date	Field Name= brand	Field Name= Manager
034	2003-01-01	Sport-1	J.F. Bush
	2003-06-01	Sport First	J.F. Bush
	2004-09-01	Sport First	G.W. Kerry
	null	End of scope	
035	2002-01-01	BigZ	X.Y. Zee
	2004-12-31	End of scope	

The time period for the first store 034 is from 2003-01-01 to 2004-09-01. The historical data given for the store replaces all existing history for that store in the OWS application from 2003-01-01. Values known before 2003-01-01, if any are unchanged.

The history has an initial set of values (Sport-1 and J.F. Bush), a change of brand (Sport First) and later a change of manager (G.W. Kerry).

For the second store 035, the time period is from 2002-01-01 to 2004-12-31. The existing historical data between the dates (2002-01-01 to 2004-12-31) is replaced, while data before and after the time period, if any is left unchanged.

Schedule Records

A schedule is related to a file or contract of an employee, for a given time period. You can define the time period by the start and end dates. A schedule is a sequence of shift items. Following are the fields of the *Schedule* record:

- *Date*
- *Activity* (or Absence or Store) code
- *StartTime*
- *EndTime*
- *PartyID* (business unit for which the activity is done)
- *WorkingHours* (value in hours for an absence)

Availability Records

Availability is a type of record that describes the work hours (preferred and fixed) of an employee according to the cyclic repetition of a standard sequence of weeks. Typically the standard sequence of weeks is as follows: *EveningWeek, DayWeek, DayWeek,* and *MorningWeek*. For each type of week, you can provide the preferred and fixed work hours of the employee, for all days of the week.

Time Window / Cycle Records

A time window describes the time slots for each day of a week. The week type is specific to each customer, for example *LowWeek, PeakWeek* and *NormalWeek*. Each time slot has a start and end time, and a value. The value represents either the number of persons or the number of hours required to complete a task.

Time window records are used to describe:

- Quantity of work to be done: Examples include the time employees take to complete a task and the number of employees required for the task.
- Time allotted to a task: Examples include the hours assigned for cleaning windows, unloading deliveries every morning before the store opens, and the opening hours of a store.

You can declare all time windows in the CCD and import the values to the database. You can also assign time windows based on a cycle indicating when the time windows apply in a time period.

Scheduling and Staffing

You can specify whether to manage each step of the weekly process (store or the department level, or shared), and the access rights to grant each user role. If the weekly process is managed at the store level, the store manager's home page displays a summary that shows the progress of each department's weekly process.

In configuring the business hierarchy, you create store and department libraries. You can define subnodes under store units (store group) and subnodes under department units (mandatory or optional) witch are referenced under store units.

For greater flexibility in meeting staffing requirements, you can lend employees to destination stores and departments on an hourly or daily basis, and the manager at the destination location can then schedule the person's activities. The application accounts for absences so that employees are not loaned to other stores at those times.

The OWS application displays loan information in the daily and weekly schedules, and employee maintenance team schedules, and maintains a history of the loans. To-do list notifications inform managers when an employee is loaned to them.

To have greater control over the conditions that permit cross-store scheduling, you can specify optimization parameters during configuration (ignore departments, strict department scheduling, and department preference).

List of Fields for Cross Store Scheduling

Field Name	Type	Description
Level	String	Indicates whether the loan will be daily or hourly
Date	Date	Date of loan of the employee
StartTime	Time	If the loan type is hourly, then the Start time indicates the start time of the loan
EndTime	Time	If the loan type is hourly, then the End time indicates the end time of the loan
NextDay	String	True or false
TransferHours	String	The total hours the employee is loaned
TeamNode	String	The team node to which the employee is loaned

Data Exchanges

Some dated data such as *Contact*, *Hiring*, and *Contract* have a common update behavior. You can specify the behavior for each element history in a message, by using specific attributes to alter the default behavior. The following are some options that you must consider, when importing data to OWS:

- If you do not provide a field for a given date, then the application resets the value to null. The field value after the import does not depend on the value before the import.
- If you do not provide a field for a given date, the value is left unchanged. It is not possible to reset a field to null and the field value after the import depends on the value before.
- For *Contract* only, if you do not provide a field for a given date, the value is reset to the default value.

Employee Records

All employee records have the employee number (*HRID*) as the key field. However, OWS also accepts *SSN* and *Badge* number as the key field.

PersonIdentification

The *PersonIdentification* record contains a list of fields identifying an employee. The record is valid at any point in time.

List of Fields for PersonIdentification

Field Name	Type	Description
FirstName	String	The employee's first name.
LastName	String	The employee's surname.
Title	String	Refers to the title, such as Mr, and Ms.
SSN	String	Refers to the employee's Social Security Number. Use this number instead of HRID to identify the employee, in messages.
HRID	String	Refers to the employee number for the HRMS application. Use this number to identify an employee in messages.
Badge	String	Refers to the employee's badge number. Use this number instead of HRID to identify the employee, in messages.

BirthDate	Date	The employee's date of birth.

Address

The *Address* record is related to an employee and provides the history of the employee's addresses over time.

List of Fields for Address

Field Name	Type	Description
Date	Date	Effective date of the record.
Street	String	The street where the employee resides.
City	String	The city where the employee resides.
State	String	The state or province where the employee resides.
Country	String	The employee's country of residence.
PostalCode	String	Postal code for the country.

Contact

The *Contact* record is related to an employee and provides the history of the employee's contact details.

List of Fields for Contact

Field Name	Type	Description
Date	Date	Effective date of the record.
HomePhone	String	The employee's home phone number.
CellPhone	String	The employee's cell phone number.
OfficePhone	String	The employee's office phone number.
EmergencyPhone	String	Phone number to use in case of emergency.
EmergencyContact	String	Name of the person to contact in case of emergency.
OfficeEMail	String	The employee's official e-mail address.

Hiring

The *Hiring* record is related to an employee and provides the history of the employee's hiring events.

List of Fields for Hiring

Field Name	Type	Description

Date	Date	Effective date of the record.
HireDate	Date	Hiring date indicating the first working day for the employee.

Employee File Records

All employee file records have the employee number (*HRID*) and file identification as the key field. Oracle Workforce Scheduling also accepts *SSN* and *Badge* number as the key field, apart from *HRID* to identify an employee.

When there is only one valid work period at a given time, you can set the file identification to 1. The file identification can be either relative to the employee or absolute (relative to the company).

All employee file records are dated.

Contract

The *Contract* record is related to an employee's file (work period) and provides the historical data of the employee's contract changes.

- You can predefine some fields for every employee.
- Some fields by employees receive default values at the business unit level. However, you can override these fields for an employee (only for those made possible in the CCD).

All values for the records are given in decimal hours: 1.50 would correspond to 90 minutes or 1.5 hours.

Groups of Fields

For each new record or value that has a new effective date, all values have to be given again, even if there is no change since the previous effective date. This is to ensure that the application resets some fields to null at some effective date.

The *Contract* record history has the following new features:

- Fields are grouped according to their business meaning.
- Allows external applications to generate messages dedicated to a single group of fields.
- If you have not made any changes to a group of values since the previous effective date, you are not expected to provide the group again with the same values, for an effective change of date. If you do not provide values, the application considers them the same and the values of the previous dated records are kept.

- Even if you make a change to one field in a group at an effective date, you must provide all fields of the same group again (otherwise, the application resets the missing fields to null). This message includes the attribute “missingField”. If it is set to “true” then the missing values are unchanged.
- For an effective date, if you provide a group without any field, the application resets all the fields of the group to null at that date.

List of Fields for Contract

Field Name	Type	Description
Date	Date	Effective date of the record
ContractType	String	Code of the contract type (it is listed in the CCD)
HRContract Group		
PayRate	Decimal	Hourly pay rate for the employee for this file
WorkRules Group		
MinWeeklyDuration	Decimal	Minimum weekly work duration
MaxWeeklyDuration	Decimal	Maximum weekly work duration
MinWorkingDays	Integer	Minimum number of working days in the week
MaxWorkingDays	Integer	Maximum number of working days in the week
MaxConsWorkingDays	Integer	Maximum number of consecutive working days over two weeks
MinDailyDuration	Decimal	Minimum daily work duration
MaxDailyDuration	Decimal	Maximum daily work duration
TimeBetweenShifts	Decimal	Minimum number of hours between two working shifts
MaxLateNights	Integer	Maximum number of working days that finish after the late night time limit
AbsenceAllowance	Decimal	Absence duration considered as work duration
ManagementPercentage	Decimal	Minimum percentage of hours of management activities to be performed during a working day
LateNightTimeLimit	Time	After this time limit, a working day is considered a late night
Salaried	Boolean	Boolean variable equal to True if the employee is salaried, False if hourly
ExportScheduleHoursByDepartment	Boolean	Export scheduled hours by departments for employees, such as salaried employees, whose scheduled hours should be converted as actuals
IsMinor	Boolean	Boolean variable equal to True if the employee is a minor, False if not a minor
TwoDaysOffOverEight	Boolean	Boolean variable that activates the constraint of two consecutive days off over eight days (scheduled week plus the day before)

Assignment

The *Assignment* record is related to an employee's file (work period) and provides the history of the employee's successive assignments for each business unit (team hierarchy).

List of Fields for Assignment

Field Name	Type	Description
Date	Date	Effective date of the record.
PartyID	String	Refers to the identification of the *team* business unit, to which you have assigned the employee.

Skill

The *Skill* record is related to an employee's file (work period) and provides the employee's list of skills for that contract.

- You can list skills in the order of preference.
- To indicate an employee's efficiency or productivity factor for a skill, you can add a number to a skill code. (The higher the number is, the more efficient the employee is at that skill.)
- When you do not provide the skill efficiency, the application uses the lowest value (0).

Consider the example in the following table. The employee preference is in the order -- *Management*, followed by *Cashier*, and finally *Other*. The employee efficiency for the skills *Management* and *Other* is 0, and for *Cashier* it is 100.
The employee had a single skill (*Cashier*) in January 2004, acquired 2 new skills *Management* and *Other* in June 2004 and since September 2004 the employee no longer works as a *Cashier.*

1/1/2004	Cashier
6/1/2004	Management; Cashier; 100;Other
9/1/2004	Management

List of Fields for Skill

Field Name	Type	Description
Date	Date	Effective date of the record.
Skills	String	Skills are listed and separated by semicolons.

Availability

The purpose of the *Availability* records is to provide the available, preferred, and fixed work hours of an employee, for each day in the defined time period

You provide the availablity, prefered, and fixed hours for an employee on a weekly basis. You must also define the initial sequence of weeks; OWS then repeats this initial sequence for the overall time period. For example, if you have provided the initial sequence of 4 weeks and the time period is for 12 weeks, then the sequence is repeated three times in the 12 week time period.

Each week defined in the initial sequence has a name, for example:

- EveningWeek:The employee works after 8:00 PM in this week.
- MorningWeek:The employee works before 10:00 AM in this week.
- DayWeek:The employee works after 8:00 AM in this week.

If necessary, you can repeat a week more than once in the sequence. For example, *EveningWeek DayWeek DayWeek MorningWeek*.

In this case, you do not have to provide the description of the *DayWeek* twice. The first instance in the sequence provides the full description, while the second instance has an empty description (already known).

List of Fields for Availabilities

Field Name	Type	Description
Availability	String	Name of the availability.
WeekType	String	Code of the week type (as DayWeek).
Index	Integer	Index of the day in the week.
(Availability) Start, End	Time	Start and end time of the availability.
(Preferred) Start, End	Time	Start and end time of the preferred hours.
(Fixed Hours) Start, End	Time	Start and end time of the fixed hours.

Schedule

A *Schedule* is related to a file (or contract) of an employee, for a given time period, and is defined by start and end dates. A schedule also contains a sequence of shifts. You can either import absences or generate a schedule in OWS.

- For importing absences to OWS: You must import only employee absences (except days off) and not the entire schedule from the external application. The imported absences replace all existing absences in OWS. If you provide a schedule without shifts, it is equivalent to deleting all employee absences for the time period. The activity names that are found in OWS pertain to absences such as sick leave, vacation, and others.
- For OWS generated schedules: The generated schedule comprises both activity and absences for a given scope.

List of Fields for Schedule Export and Absence Import

Field Name	Type	Description
Date	Date	The first day on which the activity starts.
Activity	String	Refers to the activity or absence code, defined in the CCD.
StartTime EndTime	Time	Refers to the start and end time of the activity.
PartyID	String	Refers to the organizational unit for which the activity is being performed.
WorkingHours	Integer	This field is only used in import of absences. It refers to the number of working hours of a day for which the absence is counted.

When you export schedules from OWS, shifts have three different kinds of information:

- ***Detail***: OWS describes the details of an activity for a specific shift.
- ***Day***: OWS describes the shift at the day level. The information that appears is aggregated by day. For example, the *StartTime* / *EndTime* specifies the start and the end time of the work day (start of first shift, end of last shift). *WorkingHours* is aggregated for the day.
- ***Week***: OWS describes the shift at the week level. The shift describes an aggregation for the week.

Business Unit Records

OWS maintains a history of records with dates for each organizational unit that has a class property (region, province, and district). These records provide information about each organizational unit.

Hierarchy Import

Hierarchy imports create all organization nodes starting from the company's headquarters. Before you import a store, you must define the parent nodes in the current or a previous message. The order of messages is important. You cannot import the child node unless you have imported the parent node.

The hierarchy of the company's organization is defined in the CCD. The starting point of the hierarchy is always the "root" for a team and business.

Field Name	Type	Description
Date	Date	Effective date of the record.
BusinessParent	String	Code of the business parent in the "business hierarchy".
TeamParent	String	Code of the business parent in the "team hierarchy".
UnitName	String	Long name of the business unit.

Time Window

A time window describes the time slots for each day of a week type. Week types are specific to each customer, and you can define them in the CCD. Some examples of week types are: *LowWeek, PeakWeek, EasterWeek,* and *NormalWeek*.

All time windows that you import from an external application to OWS can be described in the CCD. Time window records only define the slots of time for the various week types (for example, the opening hours of a store). They do not specify when the application has to apply these time slots during a given time period. You can apply time slots for week types using cycle records.

For example, if the sequence is *NormalWeek NormalWeek PeakWeek* for a 12-week time scope, the sequence of 3 weeks is cyclically repeated in the 12-week time period.

List of Fields for Time Window

Field Name	Type	Description
Name	String	Names of time windows are defined in the CCD.
Index	Integer	Index values belong to the interval [0; 6], where 0 corresponds to the first day of the week and 6 to the last day. A day that is not set in the message appears as a closed day. When you import an existing time window to OWS, the application replaces all the previous values.
Start	Time	Start time of the slot.
End	Time	End time of the slot.
Value	Decimal	Optional; the quantity associated with the slot.

Cycle

The *Cycle* record specifies the time window that applies for a week type. The start and end scope specifies the time interval for which the cycle is applicable. The scope can have more than the number of weeks specified in sequence, as the weeks are cyclically repeated.

The cycle begins with the type of week you define in the offset value. Each week type is referred to by an integer, for example *LowWeek 1*, *PeakWeek* 2, and *Normal Week 3*.
One single *CycleValue* gives the full sequence of *n* weeks. For example, the week type set - *LowWeek, PeakWeek, EasterWeek,* and *NormalWeek* is one *CycleValue* for *n* weeks, where n is an integer (1 and 2). A store can have more than one type of cycle.

List of Fields for Cycle

Field Name	Type	Description
RotaName	String	ID of the rotation defined by the record.
Offset	Integer	Starting week in the list.
CycleValue	String	An array of codes that provide the ordered list of week types for a cycle.

Workflow Notifications and To-Do Lists

A weekly to-do list displays notifications of tasks and messages the user needs to complete or read. The to-do list is accessible from the home page and links. The application generates notifications when a process runs such as import, or an action occurs that affects the schedule such as loaning an employee to another store. The messages displayed in the to-do list depend on context, primarily the organization node and user role.

Every organization node has at least one publish box, and messages are published for all ascending or descending nodes. Each type of message has a corresponding reader which is linked to one or more user roles. For example, the reader Schedule Alert has links to the user roles department manager and store manager, permitting users with those roles to view the schedule alerts.

After a user completes the to-do item, the user can change the status of the notification from open to closed, and when it no longer applies, delete the item. The to-do list displays all the items for the number of weeks defined by the configuration.

The notifications include data edited at the store level, such as:

- Employee data

- Store week type, time windows, distribution profiles, properties, and closing
- Activity properties
- Task properties
- Driver derived factors

Charts

You can use charts to describe the activities of a business unit or an employee.

Employee File Charts

You can import two types of charts for an employee contract, and specify the effective names of charts in the CCD.

Chart Category	Precision (Units)	Quantity Type	Description
Actual Daily Hours	Day	Decimal	Refers to the employee's actual number of working hours in a day.
Actual Daily Costs	Week	Decimal	Actual costs for a day.

Business Unit Charts

This section explains the categories of charts that you can import from an external application to OWS on the basis of business units.

For a given category, such as (Activity) Maximum Staffing, there could exist several charts, typically one for each activity defined in the CCD. Every time you declare an activity, you can also specify the names of the charts to:

- The names and the number of charts (at times, the numbers can go up to several hundreds) are specific to each customer, with 0 to 'n' charts for each chart type.
- The list of chart categories is fixed (around 20 categories), and does not depend on the customer's configuration of the application.

All charts are relative to a business unit (store or a department).

Chart Categories for Activities

You define the workload in OWS by activity. An activity is an aggregation of elementary tasks. Customers have their own list of activities. The following table describes the properties of an activity:

Chart Category	Precision	Quantity Type	Description
Daily budget of special fixed activities	Day/Week cyclic	Decimal	Daily number of hours budgeted for the activity.
Priority	Range	Integer	Daily priority of the activity.
Minimum duration	Range	Decimal	Daily minimum duration of the activity.
Minimum staffing	Range	Integer	Daily minimum number of persons to allocate to that activity.
Hourly Cost	Range	Decimal	Hourly Cost of an activity
Maximum staffing	Range	Integer	Daily maximum number of persons to allocate to that activity.

Chart Categories for Tasks

Customers have their own list of tasks (sub division of an activity). The following table describes the task properties:

Chart Category	Precision	Quantity Type	Description
Compression factor	Range	Decimal	Defines the maximum level that the hours for a given task can be compressed. If the total number of hours exceeds the Budget for a given day, OWS recalculates or 'compresses' the hours so that each daily total does not exceed the assigned budget for that day.
Expansion factor	Range	Decimal	Defines the maximum level that the hours for a given task can be expanded. If the total number of hours are less than the Budget for a given day, OWS recalculates or 'expands' the hours so that each daily total is not less than the assigned budget for that day.
Mix percentage	Range	Decimal	Allows you to define the percentage of labor standard expression result that is used as the task workload.

Chart Categories for Workload Computation

You can calculate the workload of a task based on a driver value, for example the number of items sold. By applying a formula on the driver value, you obtain the

"workload", which is the number of hours.

The formula to calculate the workload is as follows:

*Workload = (Driver Value * Labor Standard) + Fixed Workload*

Note: During configuration, you can implement a list of formulas.

Labor Standard

The Labor Standard can be either a multiplication or a division factor:

Multiplication Factor: The number of hours required to process one activity, for example the time taken by an employee to serve one customer.

Division Factor: The number of activities that can be processed in one hour. For example, the number of customers served in one hour by one employee.

Fixed Workload: It is the number of additional hours granted, in addition to the store budget even if there are no activities to process. For example, the effort required to open a store.

You can make some adjustments to the workload calculation:

Mix Percentage: You can apply it to a percentage, rather than using the driver value directly (for example, sales).

Threshold: Over the threshold value, the effective workload has a constant value.

The following table describes the Labor Standard properties:

Chart category	Precision	Quantity Type	Unit
Labor Standard (multiply)	Range	Decimal	Number of hours required to process one activity.
Labor Standard (divide)	Range	Decimal	Number of activities processed in one hour.
Fixed Workload	Range	Decimal	Number of hours.
Threshold	Range	Decimal	Number of hours.
Workload constant over threshold	Range	Decimal	Number of hours.
Mix Percent	Range	Decimal	Percentage.

Categories of Workload Driver Charts

You can calculate the workload based on driver values or charts, for example the number of customers. These driver charts are as follows:

Forecasted Values: OWS forecasts these values using statistical methods, or imports these values that are generated by external applications.

Actual History Charts: These values form the input for computing forecasted values by applying a trend.

Store Property and Activation: Store Property drivers are drivers that OWS does not forecast because they involve physical aspects of the store. They are not greatly affected by business factors.

An example of a store property is the surface area of a store. More than any business factor, it is the surface area that determines how many hours it will take to clean the store. Thus, the store surface can be considered as a driver value for cleaning.

An activation indicator is used to specify, for example, whether a store must be cleaned or not. (1 indicates that you must clean the store, and 0 indicates it is not necessary.)

Chart Category	Precision	Quantity Type	Unit
Actual history / forecast	Day Week	Decimal	Number of "things" by day. Number of "things" by week.
Store property	Range	Decimal	Number of "things" by day.
Activation	Day Week	Boolean	True / false by day. True / false by week.

Chart Categories for Profiles

Customers have their own list of profiles. OWS uses profiles to forecast cyclic occurrences. They are mainly used to forecast drivers.

A daily profile describes the activities that occur for every quarter hour in a day, for example the amount (in USD) obtained by the food sales every 15 minutes. A daily profile consists of 96 values, one for each quarter hour of the day.

A weekly profile describes the activities that occur each day of a week, for example the number of customers that a store receives by day. A weekly profile consists of 7 values, one for each day of the week.

A profile is always defined on a cyclic basis. The following table describes the profile values:

Chart category	Precision	Quantity Type	Unit
Daily Profile	15Minutes/Day cyclic	Decimal	Number of things.
Weekly Profile	Day/Week cyclic	Decimal	Number of things
Mix percentage	Day/Week cyclic	Decimal	Number of things.

Chart Categories for Week Types

In the CCD, you can declare several types of weeks. Here are some examples of week types:

0 – Low week

1 – Normal week

2 – High week

3 – Peak week

A week type chart is used to specify the week type for every week. The following table describes the week types values:

Chart Category	Precision	Quantity Type	Unit
Week Type	Week	Integer	Index of the week that indicates its type.

Organization and Logins

Organization

The customer company organization consists of two hierarchies: the business hierarchy and the team hierarchy. You can build different hierarchies for the business and team; for example, by creating a different number of levels for the team than the business hierarchy. Two nodes are required for both hierarchies: the company (initial node) and the stores. The stores must have a business node and a team node.

The organization structure is based on business and team classes. The classes are configured to describe the different hierarchy levels that exist for business and team hierarchies. Each hierarchy node is associated to a class.

Example:

Hierarchy Node	Bu Parent	Team Parent	Class
East	Vision	Vision	Region

Logins

Logins define the role to be used when the user connects to the application. The login also identifies the business and team nodes associated to this user.

Example:

Login / Password	Role	Business Node	Team Node
Region Manager	Corporate Profile	East	East

Exports from Oracle Workforce Scheduling

This chapter explains the two types of messages that you can export from OWS to an external application. They are Schedule and Charts.

Schedule

You can export schedules by:

- Business Unit
- Week

The fields that you can export are *Date*, *StartTime*, *EndTime*, *Activity* (or *Absence*), and *PartyID* (the organization unit for which the activity is applicable).
Note: Refer to the chapter on Data Exchanges for information on Schedule records.

Charts

You can use charts to export KPI values.