

Oracle

Human Capital Management Cloud Integrating with Oracle HCM Cloud

Release 12

This guide also applies to on-premises
implementations

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

The business names used in this documentation are fictitious, and are not intended to identify any real companies currently or previously in existence.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Contents

Preface	i
1 Introduction to Integrating with Oracle HCM Cloud	1
Integrating with Oracle HCM Cloud: Overview	1
Inbound Integrations: Explained	5
Outbound Integrations: Explained	6
2 Introduction to HCM Data Loader	9
HCM Data Loader: Overview	9
HCM Data Loader and Implementation Scenarios: Overview	10
HCM Data Loader Process: Explained	11
Supported Key Types: Explained	13
Source Keys: Explained	14
User Keys: Explained	15
Oracle Fusion GUIDs and Surrogate IDs: Explained	16
Business-Object Structure: Explained	16
FAQs for Introduction to HCM Data Loader	18
3 Data File Instructions and Delivery	19
HCM Data Loader Data and .Zip Files: Explained	19
File Line Instructions and File Discriminators: Explained	20
The METADATA Instruction: Explained	22
The MERGE Instruction: Explained	23
The DELETE Instruction: Explained	24
The SET and COMMENT Instructions: Explained	25
Generating and Using Template Files: Procedure	26
FAQs for Data File Instructions and Delivery	27

4	Preparing to Use HCM Data Loader	29
	HCM Data Loader Configuration Parameters	29
	Setting Configuration Parameters for Performance: Points to Consider	31
	HCM Data Loader Scope: Explained	34
	Defining Referenced Objects: Explained	36
	Reviewing Lists of Values: Explained	37
	Defining the Source-System Owner: Procedure	38
	Finding Business-Object Information: Explained	38
	Preparing to Load Workers: Points to Consider	39
	FAQs for Preparing to Use HCM Data Loader	40
5	Data Preparation	43
	Preparing the Source Data: Points to Consider	43
	General Data Considerations for HCM Data Loader: Explained	44
	Loading Images, Attachments, and Large Strings: Explained	45
	Supplying Source-Key Values: Points to Consider	46
	Updating Source Keys: Explained	47
	Supplying User-Key Values: Points to Consider	48
	Supplying Oracle Fusion Surrogate IDs: Examples	48
	Supplying Oracle Fusion GUIDs: Examples	49
	Managing Reserved Characters: Explained	50
	Loading and Maintaining Translated Objects: Explained	51
	Including Source-System References in Data Files: Explained	52
	Deleting Data: Explained	53
6	Loading Date-Effective Data	55
	Loading Date-Effective Objects: Overview	55
	Loading Objects with Multiple Changes Per Day: Explained	56
	Updating Date-Effective Objects That Have Future-Dated Records: Explained	58
	Retaining Attribute Values in Future-Dated Records: Examples	59
	Carrying Forward Changed Attribute Values in Retain Mode: Points to Consider	61
	Using Replace Mode to Update an Assignment: Examples	62
	Changing First Effective Start Date and Last Effective End Date: Examples	64

7	Loading Flexfield Data	67
	Loading Flexfield Data: Explained	67
	Loading Data for Descriptive Flexfields: Explained	68
	Loading Data for Descriptive Flexfields: Examples	69
	Loading Data for Extensible Flexfields: Explained	70
	Loading Data for Extensible Flexfields: Examples	71
8	Loading Setup Objects	73
	Loading Actions: Examples	73
	Loading Action Reasons: Examples	74
	Loading Banks: Explained	74
	Loading Bank Branches: Explained	75
	Loading External Bank Accounts: Explained	76
	Loading Calendar Events: Explained	77
	Loading Checklist Templates: Explained	78
	Loading Document Types: Explained	79
	Loading Document Types: Examples	80
	Loading Extended Lookup Codes: Examples	82
	Loading Legislative Data Groups: Examples	83
	Loading Name Formats: Explained	84
	Loading Person Types: Examples	86
	Loading Resource Exceptions: Examples	87

9 Loading Workers 89

Loading Workers: Overview	89
Loading New Workers: Points to Consider	90
Effective Start and End Dates for Worker Components: Explained	92
Deleting Worker Components: Explained	93
Checking for Duplicate Person Records: Explained	96
Person Numbers in Worker Objects: Explained	97
Loading External Identifiers: Explained	98
Loading Person Names: Points to Consider	98
Loading Person Images: Explained	99
Work Relationship Changes: Examples	99
Performing a Global Transfer: Explained	101
Loading Person Legislative Data: Explained	102
Loading Employment Terms: Explained	103
Automatic Calculation of FTE Values for Workers: Explained	104
FAQs for Loading Workers	104

10 Loading Worker-Related Objects 107

Loading Absence Cases: Explained	107
Loading Areas of Responsibility: Explained	107
Loading Calculation Cards: Explained	108
Loading Document Delivery Preferences: Explained	110
Loading Document Records: Explained	111
Loading User-Update Requests: Explained	112
Loading User-Update Requests: Examples	113
FAQs for Loading Worker-Related Objects	115

11 Loading Compensation Objects 117

Loading Salary Basis Records: Points to Consider	117
Loading Salary Records: Explained	118
Loading Salary Records: Examples	120
Loading Salary Range Differential Profiles and Values: Examples	122
Making Date-Effective Changes to Salary Range Differentials: Explained	122

12	Loading Work Structures	125
	Loading Work Structures: Overview	125
	Loading Grades: Explained	126
	Loading Grade Ladders: Explained	127
	Loading Grade Rates: Explained	127
	Loading Jobs: Explained	128
	Loading Job Families: Examples	129
	Loading Locations: Explained	130
	Loading Organizations: Explained	130
	Loading Positions: Explained	132
	Loading HCM Position Hierarchies: Examples	133
13	Loading Hierarchy Trees	137
	Loading Department Trees: Explained	137
	Loading Department Tree Nodes: Explained	138
	Loading Department Tree Nodes: Examples	139
	Loading Organization Trees: Explained	141
	Loading Organization Tree Nodes: Explained	142
	Loading Organization Tree Nodes: Examples	143
14	Loading Talent Objects	147
	Loading Goals, Goal Plans, and Goal Plan Sets: Overview	147
	Loading Goals: Explained	147
	Loading Goals: Examples	150
	Loading Goal Plans: Explained	151
	Loading Goal Plans: Examples	152
	Loading Goal Plan Sets: Explained	153
	Loading Goal Plan Sets: Examples	154
	Loading Talent Pools: Examples	155

15	Importing and Loading Data Using HCM Data Loader	157
	Importing and Loading Data: Points to Consider	157
	Importing and Loading Data: Procedure	158
	Import and Load Data Search Results	160
	Uploading Data Using HCM Data Loader: Worked Example	162
	Reviewing and Correcting Errors: Procedure	164
	Stopping HCM Data Loader Processing: Procedure	165
	Automating Data Loading: Overview	166
	FAQs for Importing and Loading Data Using HCM Data Loader	168
16	Data Loading Using Spreadsheets	169
	Spreadsheet Data Loaders: Explained	169
	Setting Up the Desktop Integration for Excel: Procedure	170
	Troubleshooting the Desktop Integration for Excel: Procedure	172
17	HCM Spreadsheet Data Loader	175
	Using HCM Spreadsheet Data Loader: Points to Consider	175
	Uploading Data Using HCM Spreadsheet Data Loader: How It's Processed	176
	HCM Spreadsheet Data Loader Access Requirements	178
	HCM Spreadsheet Data Loader Templates: Explained	178
	Defining Categories for Spreadsheet Templates: Procedure	180
	Creating and Editing Spreadsheet Templates: Procedure	181
	Designing Spreadsheet Templates: Explained	182
	Configuring Default Values for Spreadsheet Attributes: Explained	184
	Configuring Attribute Validation Using Value Sets: Procedure	185
	Autogenerated Attribute Values in Spreadsheet Templates: Explained	187
	Copying Spreadsheet Templates: Procedure	187
	Exporting and Importing Spreadsheet Templates: Procedure	188
	Generating Spreadsheets: Procedure	189
	Importing Data to a Spreadsheet: Procedure	189
	Managing Spreadsheet Data Sets: Procedure	191
	Uploading Data: Worked Example	192
	FAQs for HCM Spreadsheet Data Loader	194

18	Postload Processing	197
	Processes to Run After Loading Data: Explained	197
	Running Postload Processes: Points to Consider	199
	Validating Loaded Data: Explained	200
	Maintaining the Stage Tables: Procedure	201
	HCM Data Loader Extracts: Explained	202
	Copying an HCM Data Loader Extract: Procedure	202
	Configuring a Copied HCM Data Loader Extract: Procedure	203
	Submitting an HCM Data Loader Extract: Procedure	205
	HCM Data Loader Error Analysis Report	206
	Running the HCM Data Loader Error Analysis Report: Procedure	207
19	Payroll Data Loading	209
	Overview	209
	Initial Balance Loading: Explained	210
	Initial Balances: How They're Loaded	211
	Balance Batch Header and Lines Views	213
	Adjusting Multiple Balances: Procedure	215
	Adjusting Balances: Critical Choices	215
	Adjusting Balances: Worked Example	220
20	Payroll Batch Loader	223
	Overview	223
	Importing Data: Procedure	227
	Creating a Report to View Batch Line Errors: Worked Example	228
	Task Actions for Migrating Elements: Explained	233
	Migrating Objects: Procedure	236
	FAQs for Payroll Batch Loader	236
21	Payroll Batch Loader Using Workbooks	239
	Payroll Batch Statuses: Explained	239
	Creating Element Entries Using the Batch Loader: Worked Example	239
	Creating Globals Using the Batch Loader: Worked Example	242
	FAQs for Payroll Batch Loader Using Workbooks	244



22	Payroll Batch Loader File Loading	247
	Overview	247
	Automating Payroll Data Loads: Procedure	249
	File Format and Data Transformation for Payroll Batch Uploads: Critical Choices	252
	Batch Loader Formula Type	253
	Sample Formula for Batch Loader File Transformation	255
	Sample Batch Loader Formula for Multiple Tasks	256
23	Payroll Batch Loader Workbooks	259
	Balance Adjustments Workbook	259
	Balance Groups Workbook	261
	Bank Data Workbooks	263
	Costing Setups Workbook	270
	Element Entries Workbook	275
	Elements Workbook	281
	Globals Workbook	293
	Payment Methods Workbook	294
	Payroll Definitions Workbook	303
	Payroll Relationships Workbook	311
	Third-Party Organizations Workbook	318
	User-Defined Tables Workbooks	321
24	Time, Absence, and Pension Data for Payroll	337
	Overview	337
	Prerequisite Payroll Setup for Importing Time Entries: Explained	338
	Creating Elements for Time Card Entries: Procedure	340
	Importing Time Card Entries to Payroll: Procedure	343
	File Format for Importing Time Entries	344
	Integrating Absence Management with Global Payroll: Procedure	347
	Creating Payroll Elements for Processing Absences: Procedure	349
	Importing Absence Entries to Payroll: Procedure	351
	File Format for Importing Absence Entries	353
	Running the Load Benefit Batches Process: Procedure	356
	File Format for Importing Pension Deductions	357

25	HCM Extracts	361
	Managing Extracts	361
	Filtering Extracts	381
	Delivering Extracts	384
26	Flow Patterns	389
	Flow Patterns for Extracts and Reports: Overview	389
	Checklist and Flow Tasks: Explained	389
	Editing Flow Tasks: Points to Consider	389
	Flow Pattern Parameters: Explained	390
	Flow Task Start and Due Dates: Critical Choices	394
	Editing Flow Patterns: Examples	395
	Adding a BI Publisher Report to a Flow: Procedure	397
	Creating a Flow within a Flow: Worked Example	398
	FAQs for Flow Patterns	401
27	Benefits Data Extract	403
	Extracting Benefits Data Using HCM Extracts: Points to Consider	403
	Configuring for Plan Carriers: Procedure	403
	Valid Extract Names: Explained	406
	Generating and Transmitting Extracts: Procedure	407
	Custom Layout	408
28	Transferring Encrypted Data	417
	Setting up Encryption for File Transfer: Procedure	417
	Encrypting and Uploading Files Automatically: Procedure	419
	Transferring Files Automatically from HCM Extracts with PGP Encryption: Procedure	420
29	Oracle Taleo Recruiting Cloud Service Integration	423
	HCM and Oracle Taleo Recruiting Cloud Service Components: How They Work Together	423
	Importing Data From Oracle Taleo Recruiting Cloud Service: Explained	423
	Exporting Data to Oracle Taleo Recruiting Cloud Service: Explained	424
	FAQs for Oracle Taleo Recruiting Cloud Service and HCM Integration	426
30	Web Services	427
	Overview	427

Preface

This preface introduces information sources that can help you use the application.

Oracle Applications Help

Use the help icon  to access Oracle Applications Help in the application. If you don't see any help icons on your page, click the Show Help icon  in the global header. Not all pages have help icons. You can also access Oracle Applications Help at <https://fusionhelp.oracle.com>.

Using Applications Help

 **Watch:** This video tutorial shows you how to find help and use help features.

Additional Resources

- **Community:** Use [Oracle Applications Customer Connect](#) to get information from experts at Oracle, the partner community, and other users.
- **Guides and Videos:** Go to the [Oracle Help Center](#) to find guides and videos.
- **Training:** Take courses on Oracle Cloud from [Oracle University](#).

Documentation Accessibility

For information about Oracle's commitment to accessibility, see the [Oracle Accessibility Program](#).

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send e-mail to: oracle_fusion_applications_help_ww_grp@oracle.com.

1 Introduction to Integrating with Oracle HCM Cloud

Integrating with Oracle HCM Cloud: Overview

This guide describes the ways in which you can integrate applications with Oracle HCM Cloud. Most of the content describes ways of loading or extracting data.

This topic shows the scope of this guide and summarizes the contents of each chapter.

Introduction

Chapter	Title	Contents
1	Introduction	An introduction to this guide and an overview of the available approaches to data loading and extracts.

Data Loading Using HCM Data Loader

Chapter	Title	Contents
2	Introduction to HCM Data Loader	An introduction to HCM Data Loader and the data-loading process.
3	Data File Instructions and Delivery	Valid instructions to include in data files, how to deliver data files for upload, and how to generate and use business-object templates.
4	Preparing to Use HCM Data Loader	Setting configuration parameters, defining referenced objects, reviewing lists of values, and finding business-object information.
5	Data Preparation	General guidance for preparing various types of data, such as images and attachments, to ensure successful loading. Information about specifying the supported key types.
6	Loading Date-Effective Data	How to load date-effective objects successfully and achieve the required results.

Chapter	Title	Contents
7	Loading Flexfield Data	How to load data for descriptive and extensible flexfields.
8	Loading Setup Objects	How to load setup objects, such as actions, action reasons, name formats, and person types.
9	Loading Workers	How to load components of the worker object.
10	Loading Worker-Related Objects	How to load worker-related objects, such as absence case, document record, and user.
11	Loading Compensation Objects	How to load compensation objects, such as salary basis and salary.
12	Loading Work Structures	How to load work structures, such as grades, jobs, and organizations.
13	Loading Hierarchy Trees	How to load department and organization trees and tree nodes.
14	Loading Talent Objects	How to load goals, goal plans, and goal plan sets.
15	Importing and Loading Data Using HCM Data Loader	How to import and load data and correct any errors.

Data Loading Using Spreadsheet Loaders

Chapter	Title	Contents
16	Data Loading Using Spreadsheets	An overview of the available spreadsheet loaders and general setup requirements for spreadsheet usage.
17	HCM Spreadsheet Data Loader	Using HCM Spreadsheet Data Loader to import business objects to the HCM Data Loader stage tables and load them to the application tables.

Postload Processing

Chapter	Title	Contents
18	Postload Processing	Processes that you must run after loading specific business objects using either HCM Data Loader or HCM Spreadsheet Data Loader.

Loading Payroll Data

Chapter	Title	Contents
19	Payroll Data Loading	An overview of the ways in which you can load payroll data both during setup and later.
20	Payroll Batch Loader	Using the payroll batch loader tasks to load payroll business objects, such as element entries and payroll definitions, to Oracle HCM Cloud. Creating a report to view batch line errors. Using the payroll batch loader to migrate elements and formulas between environments.
21	Payroll Batch Loader Using Workbooks	Details of batch statuses and examples showing how to use tasks in the workbooks to load fast formula globals and element entries.
22	Payroll Batch Loader File Loading	Using the Load Batch from a File process, automating file uploads, and details of file formats and data transformation. Details of the Batch Loader Formula Type formula and sample formulas for using it for transformation or for multiple tasks.
23	Payroll Batch Loader Workbooks	Details of the workbooks for balance adjustments, balance groups, bank data, element entries, element information, payroll definitions, payroll relationships, third-party organizations, and user-defined tables.
24	Time, Absence, and Pension Data for Payroll	Loading third-party time entries, pension deductions, and absence information in XML format to Oracle Global Payroll.

Extracting Data

Chapter	Title	Contents
25	HCM Extracts	Using HCM Extracts to extract data from Oracle HCM Cloud, filtering those extracts, and selecting a delivery format.
26	Flow Patterns	Using flow patterns to complete extract reports and processes.
27	Benefits Data Extract	Setting up benefits plan carriers and extracting benefits enrollment information into an XML file for each carrier.

Transferring Encrypted Data

Chapter	Title	Contents
28	Transferring Encrypted Data	How to set up data encryption. How to upload and extract encrypted data.

Oracle Taleo Recruiting Cloud Service Integration

Chapter	Title	Contents
29	Oracle Taleo Recruiting Cloud Service Integration	How integration between Oracle Taleo Recruiting Cloud Service and Oracle HCM Cloud is managed.

Web Services

Chapter	Title	Contents
30	Web Services	Using web services to integrate web-based applications with Oracle HCM Cloud.

Inbound Integrations: Explained

This topic introduces ways of developing inbound integrations to Oracle HCM Cloud.

These include:

- HCM Data Loader
- HCM Spreadsheet Data Loader
- Payroll Batch Loader

HCM Data Loader

Use HCM Data Loader to load data in bulk from any source to Oracle HCM Cloud.

HCM Data Loader loads:


- Complex, hierarchical data
- Large volumes of data, such as complete system extracts
- Object history

You can use HCM Data Loader for both data migration and ongoing data maintenance. Major benefits of HCM Data Loader include support for:

- Most HCM business objects
- Incremental or partial data loading
- Four types of keys to identify records uniquely
- Configured descriptive flexfields and extensible flexfields, translation data, tree hierarchies, person images, and document-record attachments

HCM Data Loader provides:

- A comprehensive user interface for initiating data upload, monitoring progress, and reviewing errors
- A template file for each supported business-object hierarchy, which you can use as the basis for your own data files
- A user interface for reviewing the structure and attributes of every supported business-object hierarchy
- A web service that you can use to automate data upload

 **Note:** The precursor of HCM Data Loader is HCM File-Based Loader. For information, see Transitioning from HCM File-Based Loader to HCM Data Loader (2206596.1) on My Oracle Support at <https://support.oracle.com>.

HCM Spreadsheet Data Loader

Use HCM Spreadsheet Data Loader to import data to the HCM Data Loader stage tables from spreadsheets. HCM Spreadsheet Data Loader supports most of the business objects that HCM Data Loader supports. It offers both ease of use and flexibility. For example, you can configure spreadsheet templates to suit business needs by omitting attributes, changing attribute order, changing attribute labels, and adding help text.

Payroll Batch Loader

Use Payroll Batch Loader to load payroll business objects into Oracle HCM Cloud.

You deliver a .zip file to the Oracle WebCenter Content server. Payroll Batch Loader:

1. Imports the file contents to the Payroll Batch Loader stage tables
2. Loads payroll business objects from the Payroll Batch Loader stage tables to the application tables

You can automate both steps in the Payroll Batch Loader process using the Flow Actions Service web service.

Related Topics

- [HCM Data Loader: Overview](#)
- [Using HCM Spreadsheet Data Loader: Points to Consider](#)
- [Payroll Batch Loader Tasks: Explained](#)

Outbound Integrations: Explained

This topic introduces ways of developing outbound integrations from Oracle HCM Cloud.

These include:

- HCM Extracts
- Oracle Business Intelligence Publisher (Oracle BI Publisher)

HCM Extracts

The main way to retrieve data in bulk from Oracle HCM Cloud is HCM Extracts, which is a tool for generating data files and reports.

HCM Extracts has a dedicated interface for specifying the records and attributes to be extracted. You:

- Identify records for extraction using complex selection criteria.
- Define data elements in an HCM extract using fast formula database items and rules.

You manage HCM Extracts either in the Data Exchange work area or using the Checklists interface in the Payroll work area. Alternatively, you can run extracts using the Flow Actions Service web service from outside Oracle HCM Cloud. This feature enables you to automate the outbound extract as part of an overall integration flow.

HCM Extracts Output Formats

Using the built-in integration with Oracle BI Publisher, you can generate extracts in various formats, including CSV, XML, Microsoft Excel, HTML, RTF, and PDF.

You can download the raw hierarchical XML from a completed extract using the Extract Actions Service web service. Alternatively, you can download the formatted BI report (for example, the CSV file) using the BI ScheduleService. In both cases, the output is streamed in a byte array. Therefore, any orchestration mechanism must handle this output, possibly by means of wrapper code.

You can also generate an HCM Extract without specifying an output format at the design stage. Subsequently, you can generate a BI report based on the extract using either Oracle BI Publisher or the BI ReportService. This technique uses a dedicated BI data model that interfaces with the generated HCM Extracts XML.

Oracle BI Publisher

Oracle BI Publisher supports both scheduled and unplanned reporting, based on either predefined Oracle Transactional Business Intelligence analysis structures or custom data models. You can generate reports in various formats. To use Oracle BI Publisher for outbound integrations, you generate reports in a format suitable for automatic downstream processing, such as XML or CSV.

2 Introduction to HCM Data Loader

HCM Data Loader: Overview

HCM Data Loader is a powerful tool for bulk-loading and maintaining data. The data can be from any source. You can use HCM Data Loader for data migration, ongoing maintenance of HCM data, and coexistence scenarios, where core HR data is uploaded regularly. This topic introduces the main features and advantages of HCM Data Loader, which is available in both cloud and on-premises environments.

Business-Object Support

You can load business objects for most Oracle Fusion Human Capital Management (HCM) products. These products include 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.

Ease of Use

HCM Data Loader has a comprehensive user interface for loading data, monitoring progress, and reviewing any errors. It provides real-time information for all stages of its processing. HCM Data Loader's user interface also includes detailed information about the component hierarchies and attributes of supported business objects.

You can load data from either delimited data (.dat) files or spreadsheets. You can load most supported business objects using either method. You can generate business-object templates to use as the basis of your own .dat files. For a selected business-object component, the template includes every attribute, including those for configured flexfields. When using spreadsheets, you define spreadsheet templates to suit business needs and generate spreadsheets from those templates.

Performance

HCM Data Loader supports multithreaded processing, which enables you to upload complete data extracts without severe performance impacts. References among objects that are processed on separate threads are managed automatically. You can set HCM Data Loader configuration parameters to optimize processing for your environment.

You can also perform partial or incremental loads to update existing objects, thereby minimizing the related processing.

Supported Key Types

HCM Data Loader supports these key types for all business objects:

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

- User key

As user keys and source keys are supported, knowledge of Oracle Fusion internal IDs isn't required.

Business-Object Features

HCM Data Loader supports most business-object features and requirements. For example, you can upload:

- Current and historical records for date-effective objects. You determine the amount of history to load.
- End-dated, terminated, or inactive records.
- Translated attributes in multiple languages. You specify the character set of the data file on the **File Character Set** configuration parameter for HCM Data Loader.
- Descriptive flexfields and extensible flexfields.
- Hierarchical tree data, such as organization and department trees.
- Attachments and pictures.
- Data from multiple sources. You can include source-system references in uploaded data.

Automation

You can initiate HCM Data Loader using a web-service call, which enables you to automate data upload.

HCM Data Loader and Implementation Scenarios: Overview

You can implement Oracle HCM Cloud either in full or in coexistence mode. In a coexistence implementation, you use Oracle Fusion Talent Management or Oracle Fusion Compensation, but continue to use your existing HR applications. For both full and coexistence implementations, you implement Oracle HCM Cloud by performing the tasks that appear in your implementation project or for selected functional areas. This topic describes how HCM Data Loader supports these implementation types.

Full Implementations

If you're performing a full implementation of Oracle HCM Cloud, then you can use HCM Data Loader to bulk-load any existing HCM data at appropriate stages in the implementation. Typically, you load each type of data once only for this type of implementation. Following successful upload, you manage your data in Oracle HCM Cloud.

Coexistence Implementations

In a coexistence implementation, you use Oracle Fusion Talent Management or Oracle Fusion Compensation, but maintain your existing HR applications. For this type of implementation, you:

- Move talent data permanently to Oracle HCM Cloud, which becomes the application of record for talent data.

- Upload other types of data, such as person records, periodically to Oracle HCM Cloud. The source system remains the application of record for this data.

To implement an HCM coexistence scenario, for any source system, you can use HCM Data Loader for data upload. Follow the general instructions provided for HCM Data Loader. Oracle doesn't supply tools to extract data from your source systems. Work with your implementation partner or develop your own process for extracting the source data.

HCM Data Loader provides a Compensation Changes extract. You can extract assignment, bonus, and salary changes for a specified compensation run if you're using Oracle Fusion Compensation in a coexistence scenario.

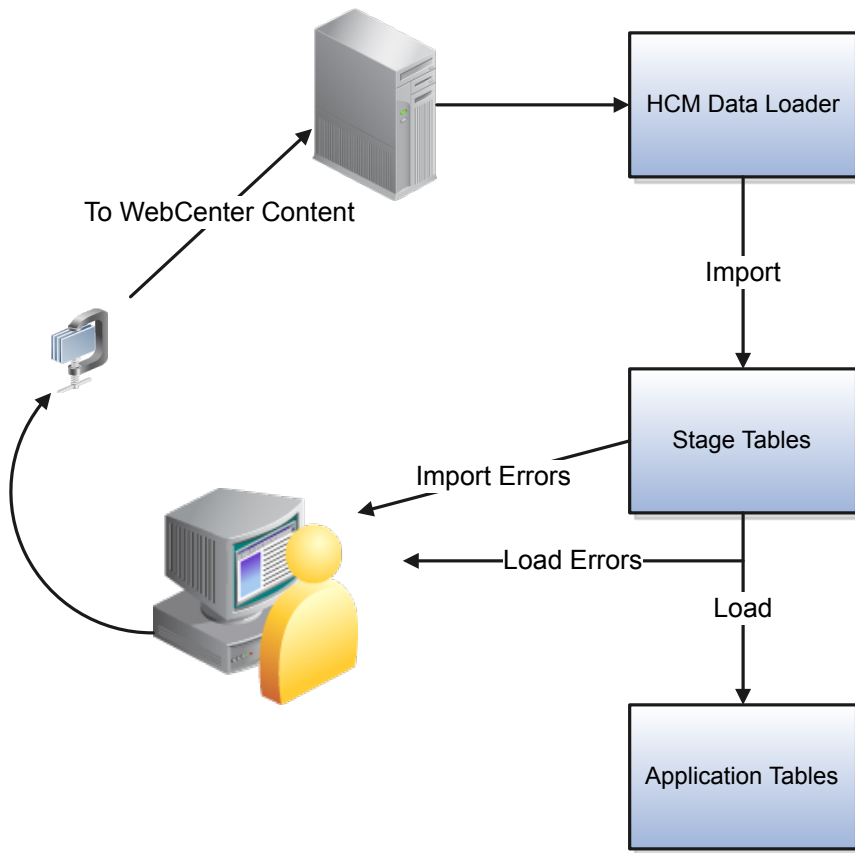
The Coexistence for HCM Feature

The Coexistence for HCM feature, which isn't available for new coexistence implementations, enables coexistence implementations for specific source environments only. The source environments supported by the Coexistence for HCM feature are Oracle PeopleSoft Enterprise Human Resources and Oracle E-Business Suite Human Resources. For information about the Coexistence for HCM feature, see the Release 11 Integrating with Oracle HCM Cloud guide on the Oracle Help Center at <http://docs.oracle.com/>.

HCM Data Loader Process: Explained

This topic outlines the process of loading data in bulk from .dat files using HCM Data Loader. The data can be from any source.


This figure provides a high-level summary of the process.



The steps of this process are:

1. You place a .zip file containing your .dat files on the Oracle WebCenter Content server.
2. You 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.
3. 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.
4. HCM Data Loader calls 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.
5. You 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.
6. You correct errors from the import and load phases in your source data.
7. You load a new .zip file containing the corrected data to the WebCenter Content server.

You repeat this process from step 2 until all of the data is successfully loaded.

 **Tip:** You can also correct load errors interactively on the HCM Data Loader interface and resubmit any corrected data from there.

Related Topics

- [Uploading Data Using HCM Spreadsheet Data Loader: How It's Processed](#)

Supported Key Types: Explained

When planning to integrate data from one system to another, you must select key types to identify records uniquely. This topic introduces the key types that HCM Data Loader supports.

What Keys Identify

In HCM Data Loader, you use unique keys to identify:

- The record that you're creating or updating.
- The parent of the record that you're creating or updating.
The parent record may either be in the same data file or exist already in Oracle HCM Cloud.
- Any objects that the record you're creating or updating references.

Supported Key Types

HCM Data Loader supports the following key types, which you can use for all types of object references. When more than one key value exists for an object, key references are resolved in this order:

1. Oracle Fusion Globally Unique Identifier (GUID)
2. Source key
3. Oracle Fusion surrogate ID
4. User key

If you supply multiple key values, then no cross-validation occurs. For example, if you supply both a GUID and a source key, then the GUID is used and the source key is ignored. If the source key identifies a different record from the GUID, then no error is raised.


 **Tip:** Whenever possible, use source keys for both creating and updating business objects.

Key-Type Features

The following table summarizes the main features of the available key types. The Create and Update columns indicate whether you can use the key type when creating and updating objects.

Key Type	Create	Update	Held on Object	Type	Generated Automatically
GUID	No	Yes	No	Hexadecimal	Yes
Surrogate ID	No	Yes (see note)	Yes	Numeric	Yes

Key Type	Create	Update	Held on Object	Type	Generated Automatically
Source Key	Yes	Yes	No	Alphanumeric	Conditionally (see note)
User Key	Yes	Yes (see note)	Yes	Alphanumeric	No

 **Note:** You can use surrogate IDs when updating objects, but the IDs may not be readily available to Oracle HCM Cloud users. Default source keys are generated only if you don't supply a source key when creating an object. You can't use user keys alone when updating some objects because their values are subject to change.

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 when the associated record is successfully created in Oracle HCM Cloud. Source keys aren't recognized in Oracle HCM Cloud until the associated record exists in Oracle HCM Cloud. Therefore, before you attempt to load data that references foreign objects, you must ensure that those foreign objects exist in Oracle HCM Cloud. For a new implementation, you're recommended to load each business object separately and ensure that it loads successfully before importing the next. If you supply all business objects in the same .zip file, then HCM Data Loader loads them in order of dependency. In this case, a reference to a foreign object fails only if the referenced object failed to load.

Source Keys: Explained

A source key is the ID that identifies a business object in its source system. Source keys are supported only for integration-enabled business objects. You can use source keys when creating or updating business objects. You're recommended to use source keys for all implementations but particularly for ongoing integrations, where data is maintained in the source system.

Source-Key Structure

A source key has these two components:

Component	Description
Source-system owner	Identifies where the data originated.
Source-system ID	ID used in the source system. The ID must be unique for the business-object component and source-system owner.

Having a source-system owner component to the key enables you to upload data from multiple source systems for the same business objects. 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 an obvious 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.

Source keys aren't held against the created record. They exist in the Integration Key Map table.

Source Keys for Date-Effective Business Objects

If you're supplying date-effective history for an object, then you must supply the source system IDs for every date-effective record in the file. The values must be the same for each line of the date-effective history.

Default Source Keys

If you provide no source key when creating an object, then the source key is created automatically. The default source-system owner is **FUSION** and the default source-system ID is the surrogate ID. You can update the source key later using the source key business object. In the source key business object, you supply the new source key and a reference to the object that you're updating.

Source-System Owner Values

The source-system owner value is validated against the HRC_SOURCE_SYSTEM_OWNER lookup. You must add your source-system names to this lookup before loading data using source-system references. Use the Manage Common Lookups task in the Setup and Maintenance work area to update HRC_SOURCE_SYSTEM_OWNER.

User Keys: Explained

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 to 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.

Oracle Fusion GUIDs and Surrogate IDs: Explained

Oracle Fusion Globally Unique Identifiers (GUIDs) and surrogate IDs are two of the four types of keys that HCM Data Loader supports. This topic describes when you can use GUIDs and surrogate IDs to identify business objects uniquely.

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.

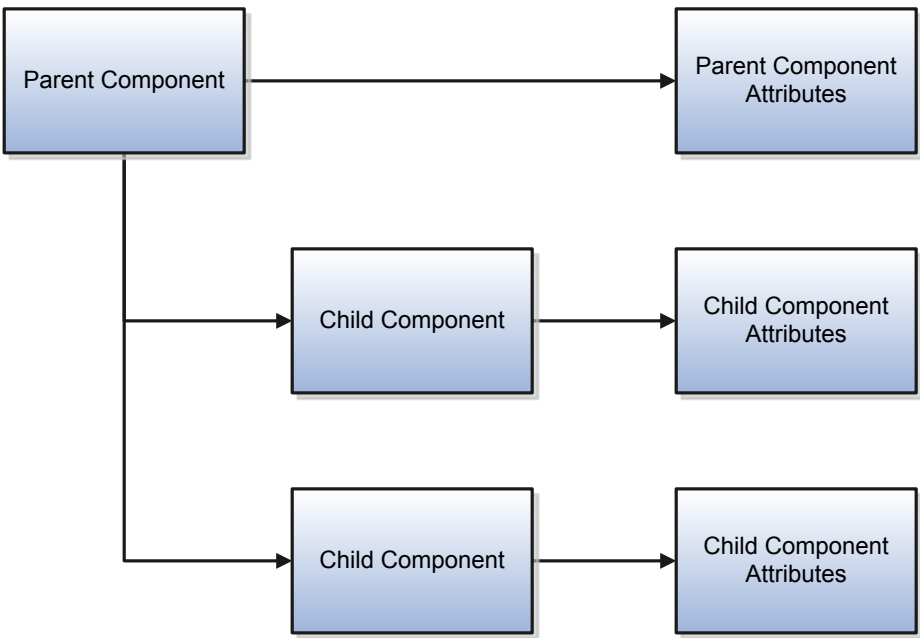
Oracle Fusion Surrogate IDs

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.

Business-Object Structure: Explained

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 rating model name, rating model code, rating level code, 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 they support neither source keys nor GUIDs.

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

Related Topics

- [Finding Business-Object Information: Explained](#)

FAQs for Introduction to HCM Data Loader

Why can't I access HCM Data Loader?

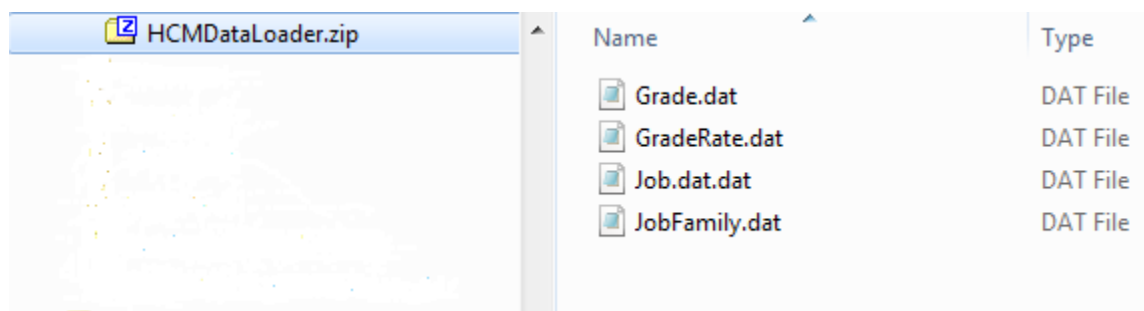
You may not have the correct permissions. You must have the Human Capital Management Integration Specialist job role or privileges to access HCM Data Loader.

3 Data File Instructions and Delivery

HCM Data Loader Data and .Zip Files: Explained

HCM Data Loader loads data from .zip files. Each .zip file can contain one or more data (.dat) files. Each data file includes data for a single business object, such as location, grade, worker, or salary. The data file has the name of a business object that is supported by HCM Data Loader, such as Job.dat or Worker.dat. The name of the data file is specified by HCM Data Loader. 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.

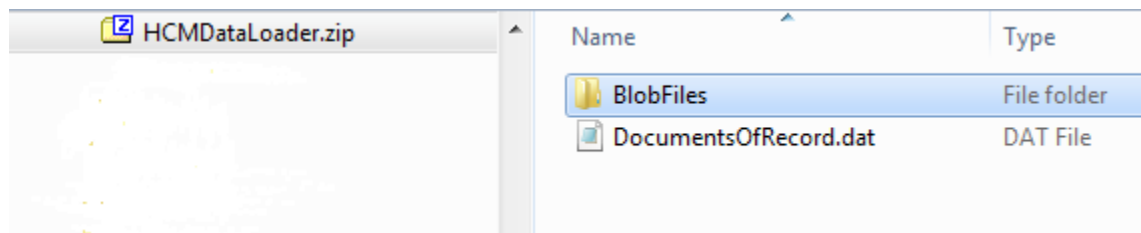
BlobFiles and ClobFiles Folders in the .Zip File

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

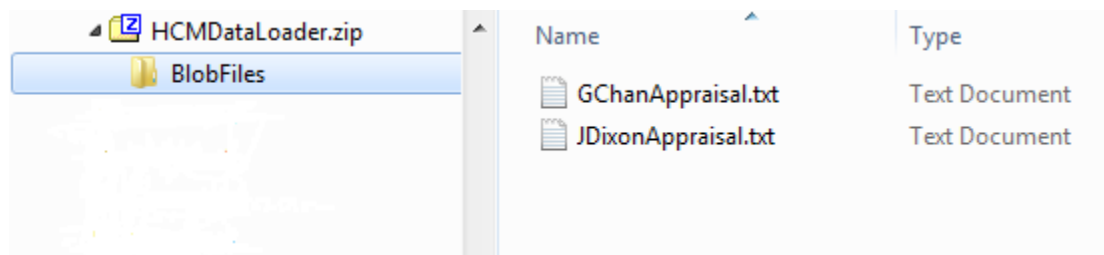
- BlobFiles, which is for binary large objects
- ClobFiles, which is for character large objects

In these folders you place files to be loaded either 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 is 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 it 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 an example .zip file structure when attachments are included.



The contents of the BlobFiles folder in this example are as shown here:



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 enables 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 Initiate Data Load page. Each template defines the complete structure of the selected object.

Related Topics

- Loading Images, Attachments, and Large Strings: Explained

File Line Instructions and File Discriminators: Explained

Each line in a data file begins with an instruction, and most lines include a file discriminator. This topic introduces the available instructions and describes the file discriminators.

File Line Instructions

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.
DELETE	Data	Identifies a business-object component to be purged from Oracle HCM Cloud.
SET	Control	<ul style="list-style-type: none">• Enables override of the reserved characters for a file.• Disables automatic postload processing of the worker object.
COMMENT	Comment	Adds a comment to the data file.

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:

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 figure shows example instructions, file discriminator, attribute names, and attributes for the job business object.

	Discriminator	Attribute Names				
Instruction		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

The METADATA Instruction: Explained

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.

The MERGE Instruction: Explained

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 record and one to update it, in the same data file. As the object 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 a worker and person name
- DELETE instructions for the person e-mail and 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.

The DELETE Instruction: Explained

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 Oracle HCM Data Loader: Business Object Documentation (2020600.1) on My Oracle Support at <https://support.oracle.com>.

Related Topics

- [Oracle HCM Data Loader: Business Object Support \(2020600.1\)](#)

The SET and COMMENT Instructions: Explained

The SET and COMMENT instructions are optional, and neither instruction includes a discriminator. This topic describes when to use the SET and COMMENT instructions.

Using the SET Instruction to Override Reserved Characters

The format of the SET instruction for overriding the reserved characters for a file is:

```
SET FILE_DELIMITER <VALUE>
SET FILE_ESCAPE <VALUE>
SET FILE_NEW_LINE <VALUE>
```

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

```
SET FILE_DELIMITER ,
```

The new value can be up to 10 characters.

The SET instruction must appear before the first METADATA line in the data file.

Using the SET Instruction to Disable Automatic Processing

You can use the SET instruction to disable some automatic postload processing of the worker object. By default, these processes run automatically after you load workers:

- Refresh Manager Hierarchy
- Update Person Search Keywords

To prevent both processes from running automatically, include this SET instruction in the Worker.dat file:

```
SET INVOKE_POST_PROCESS N
```

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

```
SET DISABLE_POST_PROCESS_TASKS RefreshManagerHierarchy
```

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

```
SET DISABLE_POST_PROCESS_TASKS UpdatePersonSearchKeywords
```

The SET instruction must appear before the first METADATA line in the data file.

Using the SET Instruction to Set the Date-Effective Maintenance Mode

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.

Using the SET Instruction to Calculate FTE Values

To request automatic calculation of full-time equivalent (FTE) values for worker assignments, include this SET instruction in the Worker.dat file:

```
SET CALCULATE_FTE Y
```

Using the COMMENT Instruction

The COMMENT instruction adds a comment to the data file. It has no effect on the processing and can appear anywhere in the file. COMMENT lines have the following format:

```
COMMENT <comment>
```

For example:

```
COMMENT Data for Business Object: Job Version: V2 Created on: 25-03-2016
```

Related Topics

- [Managing Reserved Characters: Explained](#)

Generating and Using Template Files: Procedure

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. The topic describes how to generate and use template files.


Generating a Single Template File

Follow these steps:

1. Open the Data Exchange work area and select the Initiate Data Load task on the Tasks panel tab.
By default, the Initiate Data Load page lists all business objects that HCM Data Loader supports. You can filter the list by searching for a specific object.
2. Select an object in the Search Results table and click the **Generate Template** icon on the table toolbar.
3. On the Schedule page, click **Submit**. Click **OK** to close the Confirmation dialog box.

When you click **Submit**, you start the process that identifies the business-object shape and generates the template file. You can monitor progress on the Processes tab on the Initiate Data Load page.

4. On the Business Objects tab, click the **Refresh** icon.
Once the template exists, an icon appears in the **File** column and the **Last Generated** column shows the time stamp.
5. Click the **Download File** icon to download the template.

 **Tip:** You can download an existing template. You don't have to regenerate it for each use, unless you know that the object has changed. For example, you may have configured a flexfield since the template was last generated.

Generating Multiple Template Files

Follow these steps:

1. Click the **Generate Templates** button on the Initiate Data Load page.
2. On the Schedule page, leave the **Module** field set to **All** to generate templates for all supported business objects. Alternatively, select a module. For example, set **Module** to **Talent** to generate templates for all talent business objects.
3. Once the generation process completes, you can download templates individually. Click the **Download File** icon for an object on the Business Objects tab of the Initiate Data Load page. Alternatively, follow these steps to download a single .zip file of the generated templates:
 - a. Select the Processes tab on the Initiate Data Load page.
 - b. Click the **Download File** icon for your process.

Using 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 declaring only the attributes for which you're supplying data. Remove METADATA lines and attribute names that you don't intend to use.

HCM Data Loader validates every attribute name on every METADATA line. It also identifies any potential dependencies on other business objects in the same .zip file by reviewing the attributes supplied in METADATA. For example, the Job.dat file may contain a METADATA line for the ValidGrade component with the GradeId attribute. In this case, HCM Data Loader assumes that the Job.dat file has a potential dependency on grades.

FAQs for Data File Instructions and Delivery

How should I deliver data when creating objects?

Deliver one object per .zip file. For example, create one .zip file for jobs, one for workers, and so on. Review the recommended load order on the Initiate Data Load page, and load objects in that order. Correct any errors before loading the next object to avoid data-reference errors.

How should I deliver data for incremental updates?

Supply only changed and required attribute values. Deliver all business-object files in the same .zip file. HCM Data Loader processes them in the correct order and loads referenced data before the data that references it.

4 Preparing to Use HCM Data Loader

HCM Data Loader Configuration Parameters

The HCM Data Loader configuration parameters control the HCM Data Loader import and load processes. The default settings of the parameters suit most requirements. However, you can change these settings to suit the configuration of your environment and the amount of data being loaded. Users can also override the enterprise settings when loading files either on the Schedule Request page or when using the HCM Data Loader web service.

To review or update the delivered settings for the enterprise, perform the Configure HCM Data Loader task in the Setup and Maintenance work area. This table shows the default value for each parameter.

Parameter	Default Value
Complex Error Stack Trace Occurrences by Thread	5
Data Error Stack Trace Occurrences by Thread	2
Delete Source File	Y
Enable Load Group Size Calculation	Yes
File Action	Import and load
File Character Set	UTF-8
File Encryption	None
HCM Data Loader Scope	Full
Import Cache Clear Limit	2400
Import Commit Size	100
Load Cache Clear Limit	500
Load Group Size	100
Maximum Concurrent Threads for Import	8
Maximum Concurrent Threads for Load	8

Parameter	Default Value
Maximum Percentage of Import Errors	100
Maximum Percentage of Load Errors	100
Transfer Group Size	100000

HCM Data Loader Configuration 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.

Data Error Stack Trace Occurrences by Thread

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

Delete Source File

Delete the source file from the Oracle WebCenter Content server once it's been processed. Set to N (No) to retain source files by default.

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.

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.

File Character Set

Character set used for business object and attachment files.

File Encryption

Default file encryption.

HCM Data Loader Scope

Specifies whether HCM Data Loader is used for all data loading.

From Release 10, the default value of the **HCM Data Loader Scope** parameter is **Full** for new customers only. You can't change this setting.

For existing customers upgrading from Release 9, the default value of this parameter is **Limited**. If you set it to **Full**, then you can't change the setting back to **Limited**.

Import Cache Clear Limit

Number of database rows to be processed before the cache is cleared.

Import Commit Size

Number of file lines to import between each commit.

Don't change this value unless Oracle requests the change.

Load Cache Clear Limit

Number of business-object instances to be loaded from the stage tables before the cache is cleared.

Don't change this value unless Oracle requests the change.

Load Group Size

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

Maximum Concurrent Threads for Import

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

Maximum Concurrent Threads for Load

Maximum number of threads to run concurrently when validating data or 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.

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.

Maximum Percentage of Load Errors

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

Transfer Group Size


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

Setting Configuration Parameters for Performance: Points to Consider

The HCM Data Loader configuration parameters control the HCM Data Loader import and load processes. This topic describes how to set these parameters to optimize multithreaded processing of your files. The examples in this topic provide general guidance. However, you can still load data successfully with parameter values that differ from those suggested here.

Maximum Concurrent Threads for Load


The maximum number of threads that can be used to load a business object depends on the number of servers configured in your environment. Each server supplies 8 threads. Therefore, the maximum number of threads available is 8 multiplied by the number of servers. Business objects are hosted across various servers. Therefore, you must understand not only how many servers are available but also where the business object that you're loading is hosted.

 **Tip:** The Initiate Data Load page in the Data Exchange work area shows the server that hosts each business object.

This table identifies the available servers and maximum threads in a typical environment.

Server	Number of Servers	Maximum Threads Available
Core Setup	8	64
Core Processes	2	16
Talent	1	8
Global Payroll	4	32
Workforce Management	1	8
Compensation	1	8

You can set **Maximum Concurrent Threads for Load** to the maximum threads available value. For example, if your environment has 8 Core Setup servers, then the maximum threads available is 64 (8 threads multiplied by 8 servers). Therefore, you can load objects that are hosted on the Core Setup server with the **Maximum Concurrent Threads for Load** parameter set to **64**.

 **Note:** Regardless of **Maximum Concurrent Threads for Load**, your file uses only the threads that it needs, which depends on the **Load Group Size** parameter. For example, if **Load Group Size** is set to **100**, then a file containing 500 location objects needs only 5 threads.

This table provides recommended values for the **Maximum Concurrent Threads for Load** parameter. These values are based on the number of objects that you're loading and the number of servers available for the object that you're loading.


Number of Servers	Number of Objects	Maximum Concurrent Threads for Load
1	10,000	8
2	50,000	16
5	More than 100,000	40

Load Group Size

The **Load Group Size** parameter controls the number of business objects that a thread processes at a time. Each concurrent thread processes multiple groups until all objects are processed. By default, the **Load Group Size** parameter doesn't appear on the Schedule Request page when you submit a file for processing. Instead, at the beginning of the load process, HCM Data Loader calculates the optimal load group size. This calculation is based on the available threads and the number of objects being loaded. You're recommended to leave HCM Data Loader to calculate the load group size. However, you can disable this calculation by setting the **Enable Load Group Size Calculation** configuration parameter to **No**. In this case, the **Load Group Size** parameter appears on the Schedule Request page.

If you disable automatic calculation of the load group size and instead specify a fixed value, then consider these points:

- The recommended setting for this parameter is 100, which is its default value.
- Initiating each group of objects incurs an overhead. Therefore, **Load Group Size** should not be too small.
- The statistics and status details on the Import and Load Data page are updated only at the end of every group. Therefore, **Load Group Size** should not be too large for you to monitor progress.

 **Tip:** Change this parameter from the default value only when loading smaller groups of workers. Worker is a particularly complex object with many file lines.

To optimize the bulk loading of worker objects, these **Load Group Size** values are recommended:


Number of Workers to Load	Load Group Size
More than 100,000	100
Between 50,000 and 100,000	50
Fewer than 50,000	20

Maximum Concurrent Threads for Import

The **Maximum Concurrent Threads for Import** parameter controls the number of threads available for the import phase of data loading. Set it to the same value as the **Maximum Concurrent Threads for Load** parameter. By default, both parameters are set to 8.

Processing Files in Parallel on a Single Server

Regardless of the maximum number of threads you specify when submitting a file, HCM Data Loader can use only available threads. If other activities are using threads, then HCM Data Loader processing must use fewer threads. If you're processing .zip files in parallel for objects hosted on the same server, then consider reducing the maximum concurrent threads value. This approach shares the maximum available threads among your files. Alternatively, supply the business object files in the same .zip file.

 **Note:** When you process in parallel files that contain business objects hosted on separate servers, they each use the maximum threads available.

HCM Data Loader Scope: Explained

The HCM Data Loader Scope parameter determines which business objects you can load using HCM Data Loader and which other data loaders you can use. The HCM Data Loader Scope parameter has two possible values:

HCM Data Loader Scope Value	Description
Limited	You can use HCM Data Loader to load only those business objects that aren't supported by its predecessor, HCM File-Based Loader.
Full	You can use HCM Data Loader to load all supported business objects. You can't use HCM File Based Loader, which is disabled.

For customers who upgraded from an earlier release to Release 10, the default value of this parameter is **Limited**. While you continue to use HCM File-Based Loader, leave **HCM Data Loader Scope** set to **Limited**. However, you should plan to use HCM Data Loader in place of HCM File-Based Loader. To ensure complete business-object support when using HCM Data Loader, you set **HCM Data Loader Scope** to **Full**.

For new customers, the default value of this parameter has been **Full** since Release 10.

 **Note:** You can't change **HCM Data Loader Scope** from **Full** to **Limited**.

Restricted Objects

If **HCM Data Loader Scope** is set to **Limited**, then you can't load these objects using HCM Data Loader. You can load them using HCM Data Loader only if **HCM Data Loader Scope** is set to **Full**.

Module	Business Object	Business Object Translation
Absence Management	Person Absence Entry	
	Person Entitlement Detail	
Compensation	Salary	

Module	Business Object	Business Object Translation
Global HR	Salary Basis	
	Actions	Actions Translation
	Action Reasons	Action Reasons Translation
	Content Item	Content Item Translation, Content Item Rating Description Translation
	Content Items Relationship	
	Department Tree	
	Department Tree Node	
	Education Establishment	Education Establishment Translation
	Grade	Grade Translation, Grade Step Translation
	Grade Ladder	Grade Ladder Translation, Step Rate Translation
	Grade Rate	Grade Rate Translation
	Job	Job Translation
	Job Family	Job Family Translation
	Location	Location Translation
	Organization	Organization Translation
	Person Contact	
	Person Contact Relationship	
	Position	Position Translation
	Rating Model	Rating Model Translation, Rating Category Translation, Rating Level Translation
	Talent Profile	Talent Profile Translation
	Worker	


Module	Business Object	Business Object Translation
Global Payroll	Element Entry	

If you try to load data for a business object that isn't supported in **Limited** mode, then your whole data set fails to process.

Using HCM Spreadsheet Data Loader

The version of HCM Spreadsheet Data Loader that you can use depends on the value of the **HCM Data Loader Scope** configuration parameter. If **HCM Data Loader Scope** is set to:

- **Full**, then you can use HCM Spreadsheet Data Loader to load most objects that HCM Data Loader supports. All HCM Spreadsheet Data Loader Release 12 functions are available.
- **Limited**, then you can use the version of HCM Spreadsheet Data Loader that was available in Release 11 and earlier. This version provides a static list of predefined spreadsheets

 **Note:** Spreadsheet data loaders other than HCM Spreadsheet Data Loader are unaffected by the setting of the **HCM Data Loader Scope** configuration parameter. For example, you can use the payroll batch loader regardless of the value of **HCM Data Loader Scope**.

Defining Referenced Objects: Explained

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 and the tasks that you use to review or create them. You perform these tasks in the Setup and Maintenance work area.

Business Object	Task
Application Reference Data Set	Manage Reference Data Sets
Assignment Status Type	Manage Assignment Status
Business Unit	Manage Business Unit
Content Type	Manage Profile Content Types
Currency Code	Manage Currencies
Element Type	Manage Elements
Legal Entity	Manage Legal Entity

Business Object	Task
Official Language Code	Manage Languages
Profile Type	Manage Profile Types
Schedule	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, 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 Initiate Data Load page.

Reviewing Lists of Values: Explained

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, search for relevant tasks by entering **Manage % Lookups** in the Setup and Maintenance work area. All available lookup-type tasks are listed. For example, to manage person lookup types, select 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.

Related Topics

- [Lookups: Explained](#)

Defining the Source-System Owner: Procedure

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.

Follow these steps:

1. In the Setup and Maintenance work area, search for and select the Manage Common Lookups task.
2. On the Manage Common Lookups page, search for the lookup type **HRC_SOURCE_SYSTEM_OWNER**.
3. In the search results, select the lookup type to display its lookup codes.
4. In the Lookup Codes section of the page, click the **New** icon to add a lookup-code row.
5. 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.
6. Repeat steps 4 and 5 for additional source-system owner values.
7. Click **Save and Close**.

Related Topics

- [Source Keys: Explained](#)

Finding Business-Object Information: Explained

To load a business object or component successfully using HCM Data Loader, you must understand its structure, supported key systems, data formats, and required attributes. You must also understand any business rules that apply to the object. This topic identifies where you can find this information.

Business-Object Details


On the Initiate Data Load page, you can generate data-file templates. When you click the name of a business object in the **Business Object** column in the Search Results table, you access information about the object. This information includes:

- The component hierarchy of the object
- Component details for the currently selected component in the hierarchy. These include:
 - Its parent component, file discriminator, data-file name, and the name of any translation object

 **Tip:** Click the name of the translation object to see its details.

- Whether it supports integration keys
- Whether it's date effective and, if so, whether you can change the component's first start date and last end date

- The descriptive flexfields that you can upload for the component
- For each attribute of the component, you can see:
 - Its name, type, and length
 - Whether it's required
 - Whether it's a key value or can be used as a key value and, if so, the key type
 - The names of any alternate user key attributes, which you can use in place of surrogate IDs

 **Tip:** Select **Actions - Export to Excel** to export the attribute details to a spreadsheet.

Business Rules

Business-rules information exists in Oracle Applications Help and in the Integrating with Oracle HCM Cloud guide for these business objects:

- Setup objects
- Work structures
- Worker and user-update requests
- Goals, goal plans, and goal plan sets

For objects that aren't listed here, see HCM Data Loader: Business Object Documentation (2020600.1) on My Oracle Support at <https://support.oracle.com>. This document also provides attribute descriptions and example data (.dat) files for supported business objects.

Other documents on My Oracle Support provide information about business objects for specific localizations. Many are linked from the document All White Papers for Oracle Fusion HCM (1504483.1). For example:

- Oracle Fusion HCM: HCM Address Validation (2140848.1)
- Oracle Fusion HCM: Person Name Validation (2146270.1)
- Validating National Identifier Numbers in Oracle Fusion Human Capital Management (2159758.1)

Related Topics

- [HCM Data Loader: Business Object Documentation \(2020600.1\)](#)

Preparing to Load Workers: Points to Consider

Before you load worker objects, you must decide:

- Whether you want user accounts to be created automatically for those workers.
- Whether user credentials are to be sent automatically to new users.
- Which roles are to be provisioned automatically to new users. User accounts created without at least one role are automatically suspended.

This topic describes each of these decisions.

Creating User Accounts Automatically

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

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

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

Sending User Credentials to New Users

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


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

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

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

Provisioning Roles to New Users

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

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

Related Topics

- [Role Mappings: Explained](#)

FAQs for Preparing to Use HCM Data Loader

How can I find out where a business object is hosted?

See the Initiate Data Load page in the Data Exchange work area. This page shows the module for each business object. The module equates to the server, as shown in this table.

Module	Server
Absence Management	Workforce Management
Compensation	Compensation
Global HR - Core HR	Core Processes
Global HR - Core Setup	Core Setup
Global Payroll	Payroll
Talent	Talent
Workforce Management	Workforce Management

Can I load data in multiple languages?

Yes. Language packs for the required languages must be installed. For cloud implementations, requested language packs may be installed before the environment is delivered. Alternatively, you can request language packs by raising a service request (SR).

5 Data Preparation

Preparing the Source Data: Points to Consider

This topic describes some ways in which you can prepare your data for loading to Oracle HCM Cloud and ensure that it loads successfully.

Reviewing and Cleansing the 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.

Preparing the source data in this way minimizes the problems that can occur when you upload data to Oracle HCM Cloud. It also makes it less likely that you load inaccurate data to the new environment.

Extracting Data from the Source System

You must define mappings between your source data and the Oracle HCM Cloud business-object model by comparing source and target attributes. You can review the structure of business objects in Oracle HCM Cloud by:

1. Selecting **Navigator - Data Exchange - Initiate Data Load** to open the Initiate Data Load page.
2. Selecting the name of a business object in the **Business Object** column of the Search Results table.

A Business Object Details page for the selected object opens. It shows:

- a. The component hierarchy of the object
- b. For a selected component, some general details, such as whether it's date effective, its attributes, and whether you can load flexfield data.

You must also 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

HCM Data Loader validates your data during both the import and load phases of data loading. The Data File Validator tool, available from My Oracle Support, can perform most of the data-formatting validations before you attempt to load the data. You run this utility 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.

You can download the Data File Validator tool from the My Oracle Support document, Data File Validator Tool for HCM Data Loader (document ID 2022617.1).

Related Topics

- [Data File Validator Tool for HCM Data Loader \(2022617.1\)](#)

General Data Considerations for HCM Data Loader: Explained


This topic describes some general aspects of data preparation. Follow these rules to ensure successful data upload.

Retaining Current Values

When you update existing data in Oracle HCM Cloud, you supply:

- A unique identifier for the record being updated
- 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

For nonflexfield 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 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 is uploading the data.

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

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

The expected formats for date and time values are:

Date or Time	Format
Date	YYYY/MM/DD
Time	YYYY/MM/DD HH24:MI:SS

For example: 2016/11/05 14:20:00

To set an existing date or time value to null, supply the #NULL token as the attribute value.

Related Topics

- [Loading Flexfield Data: Explained](#)

Loading Images, Attachments, and Large Strings: Explained

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.

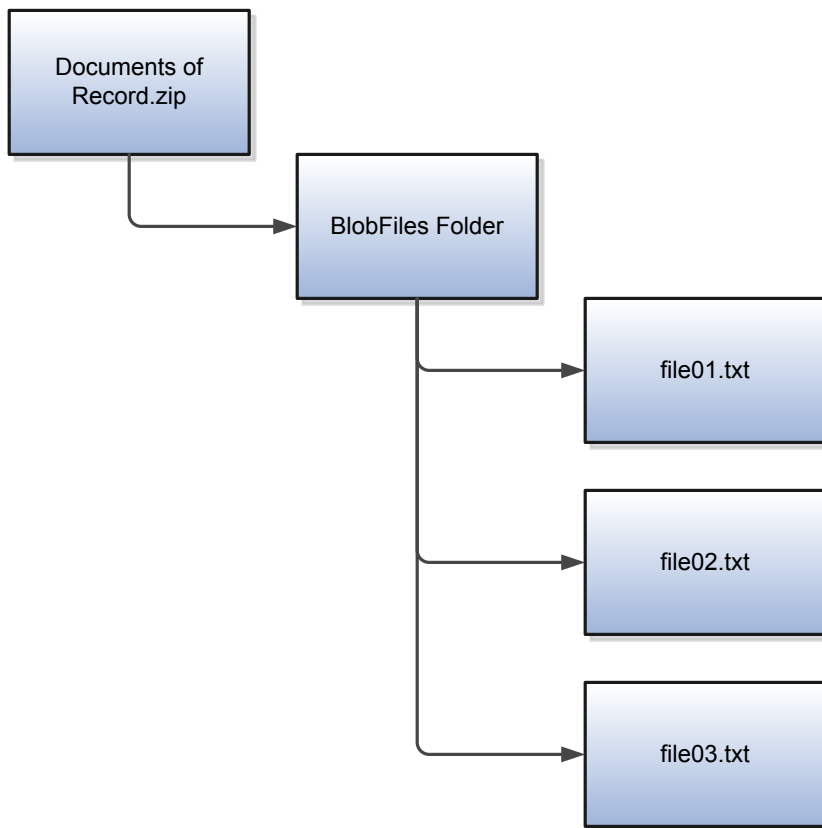
```
METADATA|DocumentAttachment|DocumentType|File|PersonNumber|...  
MERGE|DocumentAttachment|Drivers License|file01.txt|23901|...  
MERGE|DocumentAttachment|Drivers License|file02.txt|64235|...
```

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.

Supplying Source-Key Values: Points to Consider

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.

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.

Related Topics

- [Source Keys: Explained](#)
- [Defining the Source-System Owner: Procedure](#)

Updating Source Keys: Explained

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. This topic describes how to update source keys.

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
```

Supplying User-Key Values: Points to Consider

User keys are specific to the business object component that you're loading. The integrated business-object documentation 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.

Related Topics

- [User Keys: Explained](#)

Supplying Oracle Fusion Surrogate IDs: Examples

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.

Specifying 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
```

Specifying 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
```

Related Topics

- [Oracle Fusion GUIDs and Surrogate IDs: Explained](#)

Supplying Oracle Fusion GUIDs: Examples

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.

Specifying 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
```

Specifying 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.

Related Topics

- [Oracle Fusion GUIDs and Surrogate IDs: Explained](#)

Managing Reserved Characters: Explained

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,TheSteadying\newlineKier Allan
```

Related Topics

- [The SET and COMMENT Instructions: Explained](#)

Loading and Maintaining Translated Objects: Explained

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 follows:

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

Language	Source Language	Job Name
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
```

Related Topics

- [HCM Data Loader Configuration Parameters](#)

Including Source-System References in Data Files: Explained

You can include source-system references in each data line in a file. They enable you to record the database-table name, column names, and attribute values from the source system. You can see these values on the Object Errors page for an object that fails to load. Therefore, you can easily identify the data source. Source-system references are optional. This topic describes how to construct source-system references, which are made up of a name and a value.

Source-System Names

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

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

You can specify up to 10 source-system column names in the same METADATA line using the SourceRef001 to SourceRef010 tags. For example:

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

Source-system references can appear anywhere in the METADATA line after the instruction and discriminator values.

Source-System Values

Supply the source-system values on each data line, ensuring that they appear in the order specified on the METADATA line. In data lines, you must leave the source-system database-table name 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
```

Related Topics

- [Reviewing and Correcting Errors: Procedure](#)

Deleting Data: Explained

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 e-mail component of a worker object.

When you delete a parent object, its child objects and any translation objects are also deleted. For example, to delete a grade and its child business-object 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

For detailed information about deletion support, see Oracle HCM Data Loader: Business Object Documentation (2020600.1) on My Oracle Support at <https://support.oracle.com>.

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

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 an object 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.

Related Topics

- [HCM Data Loader: Business Object Documentation \(2020600.1\)](#)

6 Loading Date-Effective Data

Loading Date-Effective Objects: Overview

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
EffectiveStartDate	The start date for the attribute values. This value is required for all date-effective records.
EffectiveEndDate	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.
EffectiveSequence	When multiple changes per day occur, this attribute identifies their order.
EffectiveLatestChange	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 top 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 top 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.


Loading Objects with Multiple Changes Per Day: Explained

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.

```
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 updating an existing object with 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|...
```

Updating Date-Effective Objects That Have Future-Dated Records: Explained

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. Such records are known as future-dated records. This topic describes how updates to future-dated records are processed in each of the maintenance modes. It also shows how to set the maintenance mode.

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.

Mode	Effective End Date Value	Description
		 Tip: 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.

Setting the Maintenance Mode

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.

Retaining Attribute Values in Future-Dated Records: Examples

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 this assignment:

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 follows:

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

Effective Start Date	Effective Sequence	Effective End Date	Action Code	Job Code	Grade	Location	Hours
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.

Carrying Forward Changed Attribute Values in Retain Mode: Points to Consider

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. This topic describes how to achieve each outcome and what to consider when deciding how to proceed.

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 follows:

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 follows:

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	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.

Using Replace Mode to Update an Assignment: Examples

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.

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 is as follows:

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.


Changing First Effective Start Date and Last Effective End Date: Examples

For some objects, you can change the first effective start date or last effective end date. This topic provides examples showing how to make these changes.

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.

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:** Review the details of an object component on the Initiate Data Load page. On the Component Details tab, identify whether you can change a component's first effective start date or last effective end date.

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

7 Loading Flexfield Data

Loading Flexfield Data: Explained

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. Once the flexfields are configured, the METADATA lines in generated template files include all attributes that are required to load the flexfield data.

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 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-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|...
```

Related Topics

- [Generating and Using Template Files: Procedure](#)

Loading Data for Descriptive Flexfields: Explained

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
```

Loading Data for Descriptive Flexfields: Examples

This topic shows how to construct 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)
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.

Loading Data for Extensible Flexfields: Explained

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|_EO1_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|_EO1_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.

Loading Data for Extensible Flexfields: Examples

This topic 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)
```


8 Loading Setup Objects

Loading Actions: Examples

Actions classify changes to data such as employment and compensation records. When you create or update a record, the action value specifies the reason for the change. For example, the associated action for an assignment change may be promotion or transfer. Action reasons, which are optional, provide additional information about a change. The action reason usage component of an action specifies the relationship between an action and an existing action reason. This topic shows how to load actions and their action reason usage components using HCM Data Loader.

Creating Actions

This example Actions.dat file creates an action and associates it with an action reason usage. Both components are identified by source keys.

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

Loading Translated Action Names and Descriptions

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

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

Deleting Actions

You can delete actions and action reason usages using HCM Data Loader. This example Actions.dat file deletes an action reason usage. It must include a reference to the action to which the action reason usage belongs. The action reason usage is identified by its source key.

```
METADATA|ActionReasonUsage|SourceSystemOwner|SourceSystemId|ActionId(SourceSystemId)
DELETE|ActionReasonUsage|VISION|ARU_VISION_PROM_PERF|ACT_VISION_PROM
```

 **Note:** You can't use user keys when deleting individual action reason usages.

When you delete an action, you must also delete its action reason usages explicitly. This example Actions.dat file deletes an action and its action reason usages. All DELETE instructions are in the same file. The action and its action reason usages are identified by source keys.

```
METADATA|Actions|SourceSystemOwner|SourceSystemId
DELETE|Actions|VISION|ACT_VISION_PROM
METADATA|ActionReasonUsage|SourceSystemOwner|SourceSystemId|ActionId(SourceSystemId)
DELETE|ActionReasonUsage|VISION|ARU_VISION_PROM_PERF|ACT_VISION_PROM
```

Loading Action Reasons: Examples

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

Creating Action Reasons

This example ActionReasons.dat file creates two action reasons. The action reasons are identified by source keys.

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

Loading Translated Action-Reason Names

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

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

Deleting Action Reasons

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

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

Loading Banks: Explained

A bank is a financial institution that can have multiple branches. External bank accounts are associated with bank branches. Bank, bank branch, and external bank account are business objects whose details are stored in non-HCM tables. However,

you can load them using HCM Data Loader. This topic describes aspects of the bank object that you must understand to load banks successfully.

Loading Banks

You load bank data in a Bank.dat file for processing by HCM Data Loader. To provide a unique reference to a bank when you create it, you must supply either the **Bank Name** or the **Bank Number** attribute. You must also supply the **Country Code** attribute.

This example Bank.dat file creates a bank record using the **Bank Name** and **Country Code** user keys. It also supplies the **Bank Number** attribute.

```
METADATA|Bank|BankName|BankNumber|CountryCode
MERGE|Bank|Vision Bank|100001A|CA
```

Deleting Banks

You can't delete bank objects using HCM Data Loader. However, you can inactivate banks that are either no longer required or were entered in error. This feature, which isn't available on the Manage Banks pages, enables you to maintain an audit trail.

This example Bank.dat file provides an end date for a bank record to inactivate it. To identify an existing bank uniquely, you can use the **Country Code** and either the **Bank Name** or the **Bank Number**.

```
METADATA|Bank|BankNumber|CountryCode|EndDate
MERGE|Bank|100001A|CA|2016/03/01
```

Loading Bank Branches: Explained

A bank is a financial institution that can have multiple branches. External bank accounts are associated with bank branches. Bank, bank branch, and external bank account are business objects whose details are stored in non-HCM tables. However, you can load them using HCM Data Loader. This topic describes aspects of the bank branch object that you must understand to load bank branches successfully.

Loading Bank Branches

Bank branch data must be supplied in a BankBranch.dat file for processing by HCM Data Loader. The **Bank Branch Number** attribute, for which the field name on the Manage Bank Branches page varies by country, uniquely identifies the bank branch. It may be a required attribute, depending on country-specific validations.

You must supply one of the following combinations of attributes to provide a unique reference to the bank branch and the associated bank:

- **Bank Branch Name, Country Code**, and either **Bank Name** or **Bank Number**
- **Bank Branch Number, Country Code**, and either **Bank Name** or **Bank Number**

This example BankBranch.dat file creates a bank branch using the **Bank Branch Number, Bank Name**, and **Country Code** user keys.

```
METADATA|BankBranch|BankBranchNumber|BankBranchName|BankName|CountryCode|EftSwiftCode
MERGE|BankBranch|111111A|Toronto 3|Vision Bank|CA|12345678
```

Deleting Bank Branches

You can't delete bank branches using HCM Data Loader. However, you can inactivate bank branches that are no longer required or were entered in error. This feature, which isn't available on the Manage Bank Branches pages, enables you to maintain an audit trail.

This example BankBranch.dat file provides an end date to inactivate a bank branch.

```
METADATA|BankBranch|BankBranchNumber|BankName|CountryCode|EndDate
MERGE|BankBranch|111111A|Vision Bank|CA|2015/04/01
```

Loading External Bank Accounts: Explained

An external bank account record holds the details of a bank account at a bank branch. Bank accounts are used by payment methods to make payments to a person. Bank, bank branch, and external bank account are business objects whose details are stored in non-HCM tables. However, you can load them using HCM Data Loader. This topic describes aspects of the external bank account object that you must understand to load external bank accounts successfully.

Bank and Bank Branch

Before you can create an external bank account, you must create both the bank and the bank branch where the bank account is held.

Loading External Bank Accounts

You supply external bank account data in the ExternalBankAccount.dat file for processing by HCM Data Loader. You must supply a unique reference to the external bank account using one of the following combinations of attributes:

- **Account Number, Country Code, Currency Code, Bank Name**, and either **Bank Branch Name** or **Bank Branch Number**
- **Account Number, Country Code, Currency Code, Bank Number**, and either **Bank Branch Name** or **Bank Branch Number**

The external bank account owner component identifies a person to whom the account belongs. A bank account can be associated with multiple owners, of whom one must be the primary owner. Use the discriminator ExternalBankAccountOwner to load external bank account owner components. Set the **PrimaryFlag** attribute of the relevant component to **Y** to identify the primary owner. You must supply a unique reference to the external bank account owner using one of the following combinations of attributes:

- **Account Number, Person Number, Country Code, Currency Code, Bank Name**, and either **Bank Branch Name** or **Bank Branch Number**
- **Account Number, Person Number, Country Code, Currency Code, Bank Number**, and either **Bank Branch Name** or **Bank Branch Number**

This example ExternalBankAccount.dat file creates an external bank account and associates it with two owners, one of whom is the primary owner.

```
METADATA|ExternalBankAccount|BankNumber|BankBranchNumber|CountryCode|AccountNumber|IBAN|AccountName|
AccountType|CurrencyCode
MERGE|ExternalBankAccount|100001A|111111A|CA|12345678|CA23 ANBK 3350 1234 5678 20|J and P Smith||USD
```

```
METADATA|ExternalBankAccountOwner|BankNumber|BankBranchNumber|CountryCode|AccountNumber|CurrencyCode|
PersonNumber|PrimaryFlag
MERGE|ExternalBankAccountOwner|100001A|111111A|CA|12345678|USD|121011|Y
MERGE|ExternalBankAccountOwner|100001A|111111A|CA|12345678|USD|126231|N
```

Deleting External Bank Accounts

You can't delete an external bank account record using HCM Data Loader. However, you can make the bank account inactive. This example ExternalBankAccount.dat file makes an external bank account inactive by including the **InactiveFlag** attribute and setting it to **Y**.

```
METADATA|ExternalBankAccount|BankNumber|BranchNumber|CountryCode|CurrencyCode|AccountNumber|InactiveFlag
MERGE|ExternalBankAccount|100001A|111111A|CA|USD|12345678|Y
```

Loading Calendar Events: Explained

A calendar event is an event, such as a public holiday or plant closure, on which working time may be affected. Calendar events are optional. You specify the geographical or organizational hierarchy to which workers affected by the calendar event must belong. The hierarchy must exist before you load the associated calendar events. This topic describes aspects of calendar events that you must understand to load them successfully using HCM Data Loader. It also provides examples showing how to load calendar events.

Calendar Event Lookup Categories

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

Calendar Event Coverage

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

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

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

Creating Calendar Events

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

```
METADATA|CalendarEvent|Name|Description|Category|CoverageType|ShortCode|StartDateTime|EndDateTime|TreeCode|
TreeStructureCode|TreeVersionName|HalfDayForElapsed
```

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

Deleting Calendar Events

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

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

Loading Checklist Templates: Explained

You use a checklist template to define a sequence of related tasks with multiple performers, such as those for onboarding a new worker. For example, you can create a checklist template for new hires, with tasks for providing system access, issuing badges, allocating parking space, and so on. This topic describes how to create and maintain Checklist Template objects and the associated Checklist Template Task components using HCM Data Loader.

Checklist Categories and Task Names

You supply Checklist Template and Checklist Template Task components in a Checklist.dat file. The checklist task names must be unique for the checklist. The checklist category must exist in the CHECKLIST_CATEGORY lookup for the target environment. Use the Manage Checklist Lookups task in the Setup and Maintenance work area. The combination of checklist name and checklist category must be unique.

Checklist Actions, Areas of Responsibility, and Eligibility Profiles

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

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

Value	Meaning
ORA_LN_MGR	Line manager
ORA_WORKER	Worker
ORA_INITIATOR	Initiator

The default performer is the line manager.

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

Loading Translated Checklist Values

You can translate:

- The checklist template name and description
- Checklist task names, descriptions, and action URLs

Supply these values in the language of the user who's loading them. You supply ChecklistTranslation.dat and ChecklistTaskTranslation.dat files to translate these values into other enabled languages once the checklists and tasks exist.

Deleting Checklist Templates

You can delete Checklist Template and Checklist Template Task components using HCM Data Loader.

Loading Document Types: Explained

A document type, such as Performance Improvement Plan or Letter of Recognition, defines the purpose and treatment of documents of that type. When defining a document type, you include attributes, such as the document name, dates, and issuing authority, and specify whether they're required. You can also specify an expiration notification period, indicate whether approvals are required, and enable multiple occurrences of the document. This topic describes aspects of the Document Type object that you must understand to load document types successfully using HCM Data Loader.

Document Categories and Subcategories

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

Before loading document types, ensure that referenced document categories and subcategories exist in the target environment. To create document categories, use the Manage Document Lookups task. To create document subcategories, use the Manage Extended Lookup Codes task. Perform both tasks in the Setup and Maintenance work area.

Document Types

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

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

Document Type Delivery Preferences

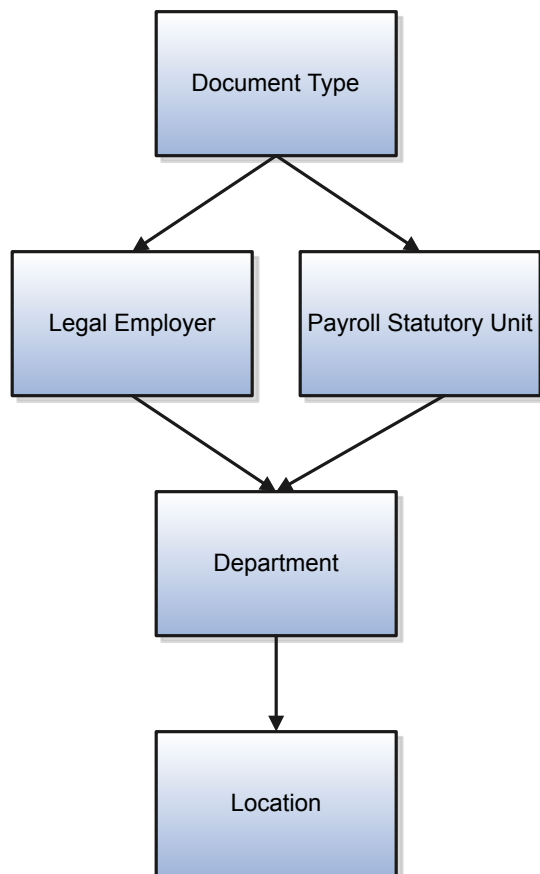
The Document Type Delivery Preference component enables users to specify delivery preferences for documents of the associated type. For example, you could enable users to specify that payslips can be delivered both on paper and online.

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

- At the payroll statutory unit level, for documents in the PAYROLL category
- At the legal employer level, for documents in all other categories

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

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



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

Related Topics

- [Document Delivery Preferences: Explained](#)
- [Document Types and Categories: Explained](#)

Loading Document Types: Examples

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

Creating Document Types Using Source Keys


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

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

Creating Document Types Using User Keys

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

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

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

Creating Document Types with Delivery Preferences

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

```
METADATA|DocumentType|SourceSystemOwner|SourceSystemId|DocumentType|Description|LegislationCode|CategoryCode|
AuthorizationRequiredFlag|MultipleOccurrencesFlag|ActiveInactiveFlag|PublishRequiredFlag|HierarchyCode|
DocumentNameRequired|DocumentNumberRequired|DateFromRequired|DateToRequired|IssuingCountryRequired|
IssuingLocationRequired|IssuingAuthorityRequired|IssuedDateRequired|CommentsRequired
MERGE|DocumentType|VISION|DTYPETest003|RS US DocType1|RS US DocType1 Description|US|AUDIT|Y|Y|N|Y|PAYROLL|R|R|
Y|Y|Y|Y|Y|Y|Y|Y|Y
METADATA|DeliveryPreference|SourceSystemOwner|SourceSystemId|DocumentTypeId(SourceSystemId)|
InitialConsentValueFlag|OnlineConsentRequiredFlag|OnlineEnabledFlag|PaperEnabledFlag|AllowWorkerOverrideFlag|
EmailEnabledFlag|LegislationCode|LevelCodeName|DocumentTypeCountry|DocumentTypeLegislationCode|
PayrollStatutoryUnitName|DepartmentName
```

```
MERGE|DeliveryPreference|VISION|DTYPETest003_Pref0|DTYPETest003|N|N|Y|Y|N|N|US||US|US||  
MERGE|DeliveryPreference|VISION|DTYPETest003_Pref1|DTYPETest003|N|N|Y|Y|N|N|US|Payroll Statutory Unit|US|US|  
GBI HCM Widgets USA|  
MERGE|DeliveryPreference|VISION|DTYPETest003_Pref2|DTYPETest003|N|N|Y|Y|N|N|US|Department|US|US|GBI HCM  
Widgets USA|HCM-1001-Corporate
```

Loading Translated Document Types and Descriptions

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

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

Deleting Document Types

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

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

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

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

Loading Extended Lookup Codes: Examples

You use extended lookup codes to provide subcategories for lookup codes. The associated lookup codes must already exist when you load extended lookup codes using HCM Data Loader. If the lookup codes are for specific legislations, then you can supply extended lookup codes for those legislations only. For example, if a lookup code has the +FR tag, then you can supply extended lookup codes for the FR legislation code only. This topic provides examples showing how to load extended lookup codes using HCM Data Loader.

Creating Extended Lookup Codes

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

```
METADATA|ExtendedLookupCode|SourceSystemOwner|SourceSystemId|LookupType|LookupCode|LegislationCode|  
ExtendedLookupCode|ExtendedLookupCodeName  
MERGE|ExtendedLookupCode|VISION|ELC_CONTRACT_LIMITED|CONTRACT_TYPE|5|NO|L|Limited Contract
```



```
MERGE | ExtendedLookupCode | VISION | ELC_CONTRACT_DIRECTOR | CONTRACT_TYPE | FR_CEO_MANDATE | FR | DC | Director Contract
```

Loading Translated Extended Lookup Code Names

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

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

Deleting Extended Lookup Codes

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

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

Loading Legislative Data Groups: Examples

Legislative data groups are used to partition payroll and related data. At least one legislative data group is required for each country where the enterprise operates. Each legislative data group holds a legislation code, currency, and cost allocation key flexfield. It's also associated with one or more payroll statutory units. This topic provides examples showing how to load and manage legislative data groups using HCM Data Loader.

Creating Legislative Data Groups

This example LegislativeDataGroup.dat file creates a legislative data group for the United Kingdom. It identifies the legislative data group using its source key. Codes are supplied for the currency and legislation.

```
METADATA | LegislativeDataGroup | SourceSystemOwner | SourceSystemId | Name | LegislationCode | DefaultCurrencyCode  
MERGE | LegislativeDataGroup | VISION | LDG_VI_UK | Vision UK | GB | GBP
```

This example LegislativeDataGroup.dat file creates a legislative data group for the United States and associates it with a cost allocation key flexfield.

```
METADATA | LegislativeDataGroup | Name | Territory | DefaultCurrency | StructureInstanceName  
MERGE | LegislativeDataGroup | Vision US | United States | US Dollar | System Test Costing
```

Deleting Legislative Data Groups

You can delete legislative data groups using HCM Data Loader. This example LegislativeDataGroup.dat file deletes a legislative data group. It identifies the legislative data group by its source key.

```
METADATA|LegislativeDataGroup|SourceSystemOwner|SourceSystemId  
DELETE|LegislativeDataGroup|VISION|LDG_VI_UK
```

Loading Name Formats: Explained

A name format is a set of rules for combining individual name components, such as first name and last name, to form a complete person name. Name formats are specific to a legislation and name-format type. The name-format types are display name, list name, order name, and full name. The predefined global format is used when no format exists for a format type and legislation. This topic describes how to load name formats using HCM Data Loader.

Understanding Format Masks

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

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

Use these codes for symbols and special characters:

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

Constructing a Format Mask

The format mask must:

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

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

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

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

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

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

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

Creating Name Formats

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

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

Deleting Name Formats

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

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

Loading Person Types: Examples

System person types are predefined values that identify groups such as employees and contingent workers. You can't create, edit, or delete system person types. However, each system person type is associated with one or more user person types, which further categorize the group. You can create, edit, and delete user person types. For example, you could define Associate and Remote Worker as user person types of the Employee system person type. For any system person type, one user person type must be identified as the default value. This topic provides examples showing how to load and manage user person types using HCM Data Loader.

Creating Person Types

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

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

Loading Translated Person-Type Names

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

```
METADATA|PersonTypeTranslation|SourceSystemOwner|SourceSystemId|Language|UserPersonType
```

```
MERGE|PersonTypeTranslation|VISION|PT_EMP_OFFICER|FR|Officier
```

Deleting Person Types

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

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

Loading Resource Exceptions: Examples

A resource exception is a deviation in availability from a work schedule or schedule assignment. The resource exception defines when a resource is unavailable. For example, a worker may be attending training and therefore unavailable between specified dates. You create resource exceptions, which are optional, for specific work schedules or schedule assignments. The work schedule or schedule assignment must exist before you can create an associated resource exception. This topic provides examples showing how to load resource exceptions using HCM Data Loader.

Creating Resource Exceptions

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

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

Deleting Resource Exceptions

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

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


9 Loading Workers

Loading Workers: Overview

The worker business object includes all information related to a person and his or her work relationships. This topic describes some general considerations that apply when you load workers.

The Worker Hierarchy

Select the worker object on the Initiate Data Load page to review its component hierarchy:

Using Source Keys

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

Multiple Instances of a Component

A person can have multiple instances of some components, such as addresses, phones, and e-mails. When you load multiple instances of a component for a person, you must:

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

The Employment Model

You must understand the employment model in the legal employer to which you're loading work relationships and assignments. In general, the employment model can be three-tier, which supports employment terms, or two-tier, which doesn't. In any legal employer, the **Employment Model** option can be set to one of these values:

- 2 Tier - Single Assignment
- 2 Tier - Multiple Assignment
- 2 Tier - Single Contract - Single Assignment
- 3 Tier - Single Employment Terms - Single Assignment
- 3 Tier - Single Employment Terms - Multiple Assignment
- 3 Tier - Multiple Employment Terms - Single Assignment

- 3 Tier - Multiple Employment Terms - Multiple Assignment

For example, any attempt to load employment terms to a legal employer where a two-tier employment model is implemented fails. Also, when loading multiple work relationships or assignments for a person, you must identify which is primary using the **PrimaryFlag** attribute. A person must have only one primary work relationship at a time and only one primary assignment in each work relationship.

Defining Referenced Values

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

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

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

Worker Termination

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

Finding Example .dat Files

For examples of .dat files showing how to load workers, see HCM Data Loader: Loading Workers (document ID 2022624.1) on My Oracle Support at <https://support.oracle.com>.

Related Topics

- [Employment Model: Explained](#)
- [Source Keys: Explained](#)
- [Reviewing Lists of Values: Explained](#)
- [Finding Business-Object Information: Explained](#)
- [Position Synchronization: Explained](#)

Loading New Workers: Points to Consider


The worker object hierarchy is complex, with many components and hierarchy levels. When you create workers, especially in large volumes, you're recommended to load only the required components initially. Once the worker objects exist, you can load related objects, such as salary and element entries. This topic identifies the components that you must load when you create workers and those that you must load in some circumstances. You can load optional worker components once the worker objects exist.

Loading Required Components of the Worker Object

Supply these components of the worker object when you create workers using HCM Data Loader:

- Worker
- Person legislative data
- Person name
- Work relationship
- Employment terms, when the legal employer supports a three-tier employment model
- Assignment

Include date-effective history, as appropriate, for each worker.

 **Note:** The worker **StartDate** and **EffectiveStartDate** attribute values must both be equal to the start date of the worker's earliest work relationship.

Loading Person Legislative Data

Person legislative data isn't a required component for creating workers. However, you're recommended to include it and provide a source-key value. Otherwise, the component is created automatically with a default source-key value.

Loading Assignment Managers

Load assignment manager components to create manager relationships only after you have loaded workers successfully. This approach ensures that the worker records of referenced managers exist before you refer to them in assignment manager components.

Loading Contracts

You can supply contract details when using either of these employment models:

- Three-tier
- Two-tier, with single contract and assignment


You must supply the contract component when creating workers, if appropriate. You can't load contract components later.

Loading Users and Roles

The enterprise option **User Account Creation** controls whether user accounts are created automatically for workers loaded in bulk. However, you can prevent user accounts being created for individual workers by including the **GeneratedUserAccountFlag** attribute of the user information component and setting it to **N**. You can also specify a user name on the **UserName** attribute of the user information component. Otherwise, the user name is generated automatically in the enterprise default format. If you set one of the following attributes or attribute pairs, then an attempt is made to link an existing user account to the worker:

- **UserGUID**
- **UserName** and **UsernameMatchingFlag**
- **EmailAddress** and **EmailMatchingFlag**

Roles are provisioned to new users automatically, as specified by current role-provisioning rules. To assign roles manually to a new worker, include the person user manual roles component with the worker object.

 **Note:** You can supply the user information and person user manual roles components only for new workers and only at the time you create them. Once a worker exists, you use the user object and its user role component to manage users and roles.

If user accounts are created by default in your environment, then you can prevent user accounts being created for terminated workers. Set the enterprise option **User Account Creation for Terminated Workers** to **No**. When you load terminated workers in bulk, this setting prevents the generated user-account requests from being processed. To set this option, perform the Manage Enterprise HCM Information task in the Setup and Maintenance work area.

Effective Start and End Dates for Worker Components: Explained

Many components of the worker object are date effective. When creating workers, you must ensure that the effective dates of individual worker components don't conflict. This topic suggests an approach to setting the earliest effective start date and last effective end date for each date-effective component. You don't have to follow this guidance. However, you must ensure that effective dates are aligned for the complete worker object.

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

Component	Earliest Effective Start Date	Last Effective End Date
Worker	Earliest start date of the worker.	The end of time.
Person Address	On or after the earliest start date of the worker.	The end of time.
Person Legislative Data	Earliest start date of the worker.	The end of time.
Person Name	Earliest start date of the worker.	The end of time.

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

Deleting Worker Components: Explained

You can delete some components of the worker object but not the entire worker. This topic identifies which components of the worker hierarchy you can delete using HCM Data Loader and describes any limitations on deletion support.

This table summarizes deletion support for the components of the worker object.

Component	Deletion Supported	Notes
Worker	No	
External identifier	Yes	
Person address	Yes	You can delete nonprimary addresses but not the primary address. Identify a new primary address before you delete the current primary address.
Person citizenship	Yes	
Person delivery method	Yes	
Person drivers license	Yes	
Person e-mail	Yes	When only one e-mail exists, you can delete it. When multiple e-mails exist, you must delete all nonprimary e-mails before you can delete the primary e-mail. Alternatively, identify a new primary e-mail before you delete the current primary e-mail.
Person ethnicity	Yes	
Person image	Yes	
Person legislative data	Yes	One legislative data record must exist for a worker. You can't delete the only record. You can delete additional records.
Person name	No	
Person national identifier	Yes	
Person passport	Yes	
Person phone	Yes	When only one phone exists, you can delete it. When multiple phones exist, you must delete all nonprimary phones before you

Component	Deletion Supported	Notes
		can delete the primary phone. Alternatively, identify a new primary phone before you delete the current primary phone.
Person religion	Yes	
Person visa	Yes	
User information	No	The user information component is available only when you create workers. This component isn't available when you update workers.
Person user manual roles	No	<p>The person user manual roles component is available only when you create workers. This component isn't available when you update workers.</p> <p>To remove a single role from a worker, you use the user object rather than the worker object. To remove multiple roles from a user, you use the user role component of the user object.</p>
Work relationship	Yes	When deleting a work relationship, you must include the CancelWorkRelationshipFlag attribute with a value of Y .
Employment terms	No	
Assignment	No	
Assignment extra information	Yes	
Assignment grade steps	Yes	
Assignment manager	Yes	
Assignment work measure	Yes	
Contract	No	
Employment terms extra information	Yes	
Employment terms grade steps	Yes	
Employment terms manager	Yes	
Employment terms work measure	Yes	

Component	Deletion Supported	Notes
Worker extra information	Yes	

Checking for Duplicate Person Records: Explained

When you load person records using HCM Data Loader, you can request a check for duplicate records. In this case, an error message is raised if you try to load a duplicate record. This topic describes how to request duplicate checking for individual person records. It also describes how this option works in conjunction with the enterprise setting for duplicate checking.


Duplicate Checking for Individual Person Records

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

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

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

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

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

Enterprise Duplicate Checking

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

PersonDuplicateCheck attribute, then the current setting of **Person Creation Service Duplicate Check** is ignored for the relevant person record.

Related Topics

- [Duplicate Checks When Importing Person Records: Points to Consider](#)

Person Numbers in Worker Objects: Explained

Each person record, regardless of person type or number of work relationships, has a unique person number. Person numbers can be either generated automatically or entered manually, depending on the setting of the enterprise **Person Number Generation Method** option. This topic describes ways of providing person numbers for uploaded worker objects.

Automatically Generated Person Numbers

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

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

Manually Entered Person Numbers

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

Loading Legacy Numbers

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

Correcting Person Numbers

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

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

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

 **Note:** You're recommended not hire workers while the correction is being made.

3. Correct the person number.
4. Reinstate the original automatic number-generation method. Consider also whether the value of the **Initial Person Number** option must be updated to avoid conflict with the correction you made.

5. Run the Update Person Search Keywords process to ensure that person searches with person number as a keyword value are successful.

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

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

Loading External Identifiers: Explained

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

External Identifiers at Person or Assignment Level

The external identifier can be captured:

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

External Identifier Types

The external identifier type is defined in lookup type ORA_PER_EXT_IDENTIFIER_TYPES, which you can extend to include additional lookup codes.

Date Values

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

External Identifier Example

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

```
SET PURGE_FUTURE_CHANGES N
METADATA|ExternalIdentifier|ExternalIdentifierNumber|PersonNumber|ExternalIdentifierType|
ExternalIdentifierSequence
MERGE|ExternalIdentifier|rtuyi45678|TestPer0TALTEST_9|Third-Party Payroll ID|1
```

Loading Person Names: Points to Consider

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

Local and Global Names

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

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

Name Attributes for Specific Legislations

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

Supplying Data-Effective History

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

Related Topics

- [Oracle Fusion HCM: Person Name Validation \(2146270.1\)](#)

Loading Person Images: Explained

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

Image Type


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

Image Size

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

Work Relationship Changes: Examples

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

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

Changing the Hire Date

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

This example changes the hire date for an existing worker.

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

Terminating a Work Relationship

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

This example terminates a work relationship.

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

Correcting the Termination Date of a Work Relationship

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

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

```
SET PURGE_FUTURE_CHANGES N
METADATA|WorkRelationship|SourceSystemOwner|SourceSystemId|PersonId(SourceSystemId)|LegalEmployerName|
CorrectTerminationFlag|ActualTerminationDate
```

```
MERGE|WorkRelationship|VISION|1009_POS|1009|Cox-6-CWB|Y|2015/10/08
```

Reversing a Termination

To reverse a termination, load the work relationship record with the **ReverseTerminationFlag** attribute set to **Y**.

This example reverses the termination of a work relationship.

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

Canceling a Work Relationship

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

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

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

Changing the Primary Assignment or Work Relationship

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

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

Performing a Global Transfer: Explained

This topic explains how to perform global transfers for both primary and nonprimary work relationships using HCM Data Loader. A global transfer is a permanent transfer from one legal employer to another. It results in the termination of an existing work relationship and the creation of a new one. The termination date of the existing work relationship is one day prior to the start date of the new work relationship.

Performing a Global Transfer on a Primary Work Relationship

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

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

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

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

Performing a Global Transfer on a Nonprimary Work Relationship

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

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

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

Updating Person-Name Legislation Codes

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

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

Loading Person Legislative Data: Explained

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

Person Legislative Data Start Dates

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

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

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

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

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

Loading Employment Terms: Explained

You can load employment terms to environments where a 3-tier employment model is implemented. This topic describes how to set attribute values on employment terms and their associated assignments when you load them using HCM Data Loader.

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

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

Automatic Calculation of FTE Values for Workers: Explained

The full-time equivalent (FTE) value is the result of dividing assignment working hours by standard working hours, which are typically those of a full-time worker. For example, if the assignment working hours are 10 and the standard working hours are 40, then the FTE value is 0.25. When you edit assignment working hours in the user interface, FTE is calculated automatically. When you load assignment records using HCM Data Loader, you can specify FTE values on the Assignment Work Measure component. Alternatively, you can request that FTE values be calculated automatically. This topic describes how to request automatic calculation of FTE values.

Calculating FTE Values Automatically

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

```
SET CALCULATE_FTE Y
```


When you include this instruction, the following rules apply:

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

You may find automatic calculation of FTE values helpful if:

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

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

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

Related Topics

- [File Line Instructions and File Discriminators: Explained](#)
- [The SET and COMMENT Instructions: Explained](#)

FAQs for Loading Workers

How do I load person national identifiers?

You must include any formatting characters in the value that you specify on the **NationalIdentifierNumber** attribute of the Person National Identifier component. For example, to load the US social security number **987-65-4322**, you must include

the hyphens. Don't specify the number as **987654322**. If you omit the formatting, which is country-specific, then duplicate person records aren't found when the checking is based on national identifier.

How do I load person phones?

You must include any formatting characters in the value that you specify on the **PhoneNumber** attribute of the Person Phone component. For example, to enter the US phone **650.555.0185** in the user interface, you enter **650** in the **Area Code** field and **555-0185** in the **Number** field. When loading this number using HCM Data Loader, you must specify the value **555-0185** on the **PhoneNumber** attribute.

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

10 Loading Worker-Related Objects

Loading Absence Cases: Explained

You use absence cases to group related absences, typically those having the same cause. For example, a worker may take two periods of sickness absence after sustaining an injury at work. In this case, you can associate the two absences with the same absence case for ease of management. This topic describes how to load Absence Case objects for a worker using HCM Data Loader. Use the `AbsenceCase.dat` file

Absence Categories

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

Loading Absence Cases

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

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

Deleting Absence Cases

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

Loading Areas of Responsibility: Explained

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

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

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

Responsibility Type

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

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

The responsibility type is validated against the lookup type PER_RESPONSIBILITY_TYPES.

Responsibility Name

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

Scope of Responsibility

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

Deleting Areas of Responsibility

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

Related Topics

- [Areas of Responsibility: Explained](#)
- [Securing Person Records: Points to Consider](#)

Loading Calculation Cards: Explained

Calculation cards capture values required for payroll calculation of some earnings and deductions, such as absence payments and involuntary deductions. Various types of calculation cards exist. For example, UK calculation cards include those for statutory deductions, automatic enrollment in pensions, court orders and student loans, and benefits and pensions. Most legislations support the Calculation Card object. However, the types of calculation cards that are supported vary by legislation. This topic describes some general considerations that apply to all calculation cards when you load them using HCM Data Loader. For legislation-specific information, see All White Papers for Oracle Fusion Applications HCM (Document ID 1504483.1) on My Oracle Support at <https://support.oracle.com>.

Each calculation card has one or more components, and each component may have one or more sets of component details. For example, in the UK court orders and student loans calculation card, each component corresponds to a different type of court order. You enter information about that court order in the component details.


Calculation cards are used mainly at the level of the payroll relationship. Depending on the legislation, they can also be used at the level of the payroll statutory unit or tax reporting unit. However, HCM Data Loader supports calculation cards only at the payroll-relationship level.

Key Support

Calculation Card objects are integration-enabled. They support all key types that HCM Data Loader supports.

Using the Reconcile Integration Keys Process

Source keys are supported for integration-enabled objects only. Any business-object occurrences that were created before the Calculation Card business object was enabled for integration may not have **SourceSystemOwner** and **SourceSystemId** values. To supply source keys for these objects, run the Reconcile Integration Keys process for individual components of the Calculation Card business object. This process allocates default source keys to any instance of a component without a source key.

 **Tip:** If required, you can update these default source keys later using the Source Key object.

Run the Reconcile Integration Keys process once only for each component of the Calculation Card object. This table lists the components with their integration object names. You select the integration object name when running the Reconcile Integration Keys process.

Calculation Card Component	Integration Object Name
Calculation Card	CalculationCard
Card Association	CalculationCardAssociations
Component Association	
Card Association Detail	CalculationCardAssociationDetails
Component Association Detail	
Card Component	CalculationCardComponents
Calculation Value Definitions	ValueDefinition
Enterable Calculation Value	RangelItem
Component Detail	CalculationComponentDetails

Translating Attribute Values

The values of these attributes are translatable and vary depending on the language of the user session:


- **DirCardDefinitionName**
- **DirCardCompDefName**

- **ValueDefinitionName**

To ensure that the CalculationCard.dat file uploads correctly, the user loading the file must use the same language as the user who created it. Supply the translated values on the base object. No separate translation object exists.

Changing Logical Start and End Dates

In general, you can change the first effective start date and last effective end date of an existing calculation card. Include the **ReplaceFirstEffectiveStartDate** and **ReplaceLastEffectiveEndDate** attributes in your METADATA line, as appropriate, and set them to **Y**. Provide the new values on the **EffectiveStartDate** and **EffectiveEndDate** attributes.

 **Note:** Legislation-specific validations may mean that some changes to logical start and end dates aren't valid.

Deleting Calculation Cards

You can delete all components of the Calculation Card object using HCM Data Loader. When you delete a Calculation Card component, its child components are deleted automatically. You can also delete individual child components.

Related Topics

- [HCM Data Loader: Loading Calculation Cards for Australia \(2124649.1\)](#)
- [HCM Data Loader: Loading Calculation Cards for France \(2233630.1\)](#)
- [HCM Data Loader: Loading Calculation Cards for United Kingdom \(2098480.1\)](#)

Loading Document Delivery Preferences: Explained

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

Document Type

Before you can load document delivery preferences for a person:

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

Loading Document Delivery Preferences

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

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

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

Deleting Document Delivery Preferences

You can delete Document Record Delivery Preference objects for a person using HCM Data Loader.

Loading Document Records: Explained

Document records store information about documents, such as visas, licenses, and medical certificates, for a person. Document records can include electronic versions of the documents as attachments. This topic describes how to load Document Record and Document Record Attachment components for a person using HCM Data Loader.

Document Type

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

Attachment Files

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

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

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

Loading Document Records

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

```
METADATA|DocumentsOfRecord|SourceSystemOwner|SourceSystemId|PersonId(SourceSystemId)|DocumentCode|
DocumentName|DateFrom|DateTo|IssuingAuthority|IssuedDate|IssuingCountry
MERGE|DocumentsOfRecord|VISION|56883|134003891|PASSPORT_2442|UK Passport|2012/04/03|2022/04/03|United Kingdom
Passport Control|2012/04/03|UK
METADATA|DocumentAttachment|SourceSystemOwner|SourceSystemId|DocumentsOfRecordId(SourceSystemId)|DataTypeCode|
URLorTextorFileName|File
```

```
MERGE | DocumentAttachment | VISION | 90982 | 56883 | FILE | Passport_2442 | JSmithPassport.pdf
```

Deleting Document Records

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

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

```
METADATA | DocumentsOfRecord | SourceSystemOwner | SourceSystemId  
DELETE | DocumentsOfRecord | VISION | 56883
```

Related Topics

- [Loading Images, Attachments, and Large Strings: Explained](#)

Loading User-Update Requests: Explained

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

The User Component

You use the user component to create requests to:

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


You also use this component to update the **CredentialsEmailSent** value in the PER_USERS table. This value records whether the account and password details have been sent to the user. Valid values are **Y** (yes) and **N** (no). Once the e-mail has been sent, it can't be sent again unless you reset this value. For example, to reset user passwords, you can run the Send User Name and Password E-Mail Notifications process. Before doing so, you can set **CredentialsEmailSent** to **N** to ensure that users are notified of new passwords. Otherwise, they aren't notified.

The User Role Component

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

Creating User Accounts

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

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

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

Running Send Pending LDAP Requests

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

Related Topics

- [Send Pending LDAP Requests: Explained](#)

Loading User-Update Requests: Examples

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

Updating a User

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

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

This example creates a request to suspend an active user.

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

This example creates a request to activate a suspended user.

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

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

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

Updating User Roles

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

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

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

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

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

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

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

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

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

Creating a User Account

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

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

Creating a user account is a two-step process:

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

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

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

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

```
METADATA|User|PersonNumber|Suspended
METADATA|UserRole|PersonNumber|RoleCommonName|AddRemoveRole
```



```
MERGE|User|12312|N  
MERGE|UserRole|12312|ORA_PER_EMPLOYEE_ABSTRACT|ADD  
MERGE|UserRole|12312|ORA_PER_LINE_MANAGER_ABSTRACT|ADD
```

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

FAQs for Loading Worker-Related Objects

Can I load allocated checklists using HCM Data Loader?

Yes. Both the checklist template and the worker to whom the checklist is allocated must exist in the target environment.

11 Loading Compensation Objects

Loading Salary Basis Records: Points to Consider

A salary basis is required for all salary records. It identifies the currency and frequency of the base pay amount and specifies whether the salary includes salary components. The payroll element that you link to the salary basis is used to create element entries to pay the salary amount. If you link a grade rate to a salary basis, then the worker's salary range is derived automatically based on the worker's grade. This topic describes aspects of the Salary Basis object that you must understand to load salaries and salary bases successfully using HCM Data Loader.

Payroll Elements

The payroll element and input value that you link to a salary basis are used to pay worker salaries. The currency value of the payroll element determines the salary basis currency. Elements that you link to salary bases must exist in the target environment before you load salary bases. In addition, you must create those elements as recurring elements:

- In the same legislative data group as the salary basis
- With a primary classification of either **Standard Earnings** or **Information**
- With open eligibility, meaning that no eligibility criteria are defined at the element level

To link the same payroll element to multiple salary bases, you must configure the element to support multiple entries in the same period.

For each legal employer, you specify the level at which salary is held.

- If salary is held at the employment-terms level, then payroll elements must also be configured at the employment-terms level. In addition, you must define the payroll assigned to workers at the payroll-terms level.
- If salary is held at the assignment level, then you must configure payroll at the assignment level. You can define the payroll assigned to workers at either the payroll-terms or assignment level.

If any of these conditions for payroll elements isn't met, then your salary bases are likely to load incorrectly. For example, they may not be associated with a legislative data group or linked with a payroll element. In this case, you must correct these incorrect salary bases manually in the application. For more information about creating payroll elements, see *Configuring Payroll Elements for Use in Oracle Fusion Compensation* (1589502.1) on My Oracle Support at <https://support.oracle.com>.

Grade Rates

You can link a grade rate to a salary basis to validate worker salaries against the salary ranges defined for the grade. The ranges defined on the grade rate are used to calculate salary metrics such as compa-ratio, quartile, and quintile. These rules apply:

- The worker's grade must be included in the grade rate linked to the worker's salary basis.
- The frequency of the grade rate doesn't have to match the frequency of the salary basis. For example, you can link a grade rate where **Frequency** is **Annually** with a salary basis where **Frequency** is **Monthly**. However, the grade rate currency must match the currency of the payroll element linked to the salary basis.
- The grade rate must be defined in the same legislative data group as the salary basis.

Salary Basis Code

The **SalaryBasisCode** attribute records the frequency at which base pay is stored and displayed. Valid values, which are defined in the CMP_SALARY_BASIS lookup type, are:

- ANNUAL
- HOURLY
- MONTHLY
- PERIOD


PERIOD (payroll period) retrieves the salary frequency from the payroll defined at the payroll terms level for workers. Use this value only when payroll is defined at the payroll terms level. When Salary Basis Code is PERIOD, leave the **SalaryAnnualizationFactor** attribute blank.

Salary Basis Components

Salary basis components itemize salary adjustments. Use the **ComponentUsage** attribute of the Salary Basis object to specify whether you're loading salary basis components. Valid values are defined in the CMP_COMPONENT_USAGE lookup type and shown in this table.

Value	Meaning
NO_COMPONENT	Components aren't used and aren't loaded with the Salary Basis object.
SELECTED_COMPONENTS	Components are used and are loaded with the Salary Basis object.
USER_DECIDES_USAGE	Managers select the components to use when adjusting salary. Components aren't loaded with the Salary Basis object.

If you set the **ComponentUsage** attribute to **SELECTED_COMPONENTS**, then you load the components using the SalaryBasisComponent discriminator. If you omit the **ComponentUsage** attribute, then **NO_COMPONENT** is assumed.

 **Tip:** If you're unsure about component usage, then specify **USER_DECIDES_USAGE**. This value is the least restrictive.

Provide one Salary Basis Component record for each component type. The **AttributeValue** attribute of the Salary Basis Component record identifies the component type, such as PROMOTIONAL, COST_OF_LIVING, or ADJUSTMENT. This value is validated against the CMP_SALARY_COMPONENTS lookup type.

Deletion of Salary Basis Records

You can delete a Salary Basis object using HCM Data Loader only if it's not assigned to workers and hasn't been used to create salary records. Unless many salary bases were created in error and not assigned, you're unlikely to delete salary bases using HCM Data Loader. You can delete individual Salary Basis objects more easily in the application.

Related Topics

- [Configuring Payroll Elements for Use in Oracle Fusion Compensation](#)

Loading Salary Records: Explained

A salary record holds a worker's base pay amount for a specified period and currency. It also displays the salary range used to validate base pay and calculate metrics such as compa-ratio and quartile. Depending on how the associated salary basis is configured, salary can include zero or more components. This topic describes aspects of the Salary object that you must understand to load salary records successfully using HCM Data Loader.

Preparing to Load Salary Records

Before you can load Salary objects:

- Employment records must exist in the target environment.
- Worker assignments must include a grade, and the grade must be included in the grade rate that is linked to each worker's salary basis. These requirements must be met for salary range to appear and salary metrics to be calculated correctly on the salary record.

Workers whose salary basis has a frequency of **PERIOD** must be assigned to a payroll defined at the payroll terms level. This payroll assignment, which you can perform using the payroll batch loader, must be complete before you load salary records.

Loading Salary

Supply salary data in the Salary.dat file for processing by HCM Data Loader. If you're loading salary components:

- Set the **MultipleComponents** attribute of the Salary object to **Y**. Otherwise, set it to **N**.
- Set the **SalaryAmount** attribute of the Salary object to the base salary amount plus the sum of any salary components.

Set the **ActionCode** attribute of the Salary object to the action associated with the salary record. This table shows valid values for the **ActionCode** attribute.

Action Code	Action Name
ALLOCATE_GRP_CMP	Allocate Workforce Compensation
CHANGE_SALARY	Change Salary
GLB_TRANSFER	Global Transfer
HIRE	Hire an Employee
HIRE_ADD_WORK_RELATION	Add Employee Work Relationship
PROMOTION	Promotion
REHIRE	Rehire of Worker
TRANSFER	Transfer

Action Code	Action Name
-------------	-------------

You can create the salary at either the employment terms or the assignment level. To create the salary:

- At the employment terms level, supply a unique reference to the employment terms. Grade and full-time equivalent (FTE) are obtained from the employment terms.
- At the assignment level, supply a unique reference to the assignment. Grade and FTE are obtained from each assignment.

FTE is determined by comparing the standard working hours of the worker's legal employer with the working hours on the worker's employment terms or assignment. When the payroll element attached to the worker's salary basis has a primary classification of Standard Earnings, the annual amount sent in the element entry is adjusted for FTE. Otherwise, the amount sent in the element entry isn't adjusted for FTE.

Loading Salary Components

Set the **ComponentReasonCode** attribute. Valid values are defined in the CMP_SALARY_COMPONENTS lookup type. Use the Manage Base Pay Lookups task in the Setup and Maintenance work area to manage the lookup type. Each component reason code can be used once only for each salary record.

Specify either a **ChangeAmount** or a **ChangePercentage** for each component record. Don't specify both. You add the **ChangeAmount** to the base pay amount to derive the new salary amount for the salary record. The **ChangePercentage** value is a percentage of the base pay amount. The corresponding percentage value is calculated automatically.

Updating Existing Salary Records

When you update an open-ended salary, the existing salary is automatically end-dated on the day prior to the **DateFrom** attribute supplied for the new salary.

Loading Salary Records: Examples

This topic shows how to load and manage salary records for a worker assignment using HCM Data Loader.

Creating a Salary Using Source Keys

This example Salary.dat file creates a Salary object with components against an existing assignment. Source keys are used to identify both the Salary object and the associated assignment.

```
METADATA|Salary|SourceSystemOwner|SourceSystemId|AssignmentId(SourceSystemId)|DateFrom|SalaryAmount|
SalaryBasisName|ActionCode|SalaryApproved|MultipleComponents
MERGE|Salary|VISION|1012SAL15|1012_EMP_ASG|2015/01/01|65000|SalBasis_1007|CHANGE_SALARY|Y|N
METADATA|SalaryComponent|SourceSystemOwner|SourceSystemId|SalaryId(SourceSystemId)|DateFrom|
ComponentReasonCode|ChangeAmount|ComponentApproved
MERGE|SalaryComponent|VISION|1012SAL15_COL|1012SAL15|2015/01/01|COST_OF_LIVING|60000|Y
MERGE|SalaryComponent|VISION|1012SAL15_ADJ|1012SAL15|2015/01/01|ADJUSTMENT|5000|Y
```

Updating a Salary Using Source Keys

This example Salary.dat file updates an existing, open-ended salary record by supplying changes to the Salary object and its components. Source keys are used to identify all components of the Salary object.

```
METADATA|Salary|SourceSystemOwner|SourceSystemId|AssignmentId(SourceSystemId)|DateFrom|DateTo|SalaryAmount|
SalaryBasisName|ActionCode|SalaryApproved|MultipleComponents
MERGE|Salary|VISION|1012SAL16|1012_EMP_ASG|2016/01/01||70000|SalBasis_1007|CHANGE_SALARY|Y|N
METADATA|SalaryComponent|SourceSystemOwner|SourceSystemId|SalaryId(SourceSystemId)|DateFrom|
ComponentReasonCode|ChangeAmount|ComponentApproved
MERGE|SalaryComponent|VISION|1012SAL16_COL|1012SAL16|2016/01/01|COST_OF_LIVING|5000|Y
MERGE|SalaryComponent|VISION|1012SAL16_ADJ|1012SAL16|2016/01/01|ADJUSTMENT|0|Y
```

Creating a Salary Using User Keys

This example Salary.dat file creates a Salary object with components against an assignment. User keys are used to identify both the Salary object and the associated assignment.

```
METADATA|Salary|AssignmentNumber|DateFrom|SalaryAmount|SalaryBasisName|ActionCode|SalaryApproved|
MultipleComponents
MERGE|Salary|1014|2015/01/01|53500|SalBasis_1007|CHANGE_SALARY|Y|N
METADATA|SalaryComponent|AssignmentNumber|DateFrom|ComponentReasonCode|ChangeAmount|ComponentApproved
MERGE|SalaryComponent|1014|2015/01/01|COST_OF_LIVING|52000|Y
MERGE|SalaryComponent|1014|2015/01/01|ADJUSTMENT|1500|Y
```

Updating a Salary Using User Keys

This example Salary.dat file updates an existing, open-ended salary record by supplying changes to the Salary object and its components. User keys are used to identify all components of the Salary object.

```
METADATA|Salary|AssignmentNumber|DateFrom|DateTo|SalaryAmount|SalaryBasisName|ActionCode|SalaryApproved|
MultipleComponents
MERGE|Salary|1014|2016/01/01||53900|SalBasis_1007|CHANGE_SALARY|Y|N
METADATA|SalaryComponent|AssignmentNumber|DateFrom|ComponentReasonCode|ChangeAmount|ComponentApproved
MERGE|SalaryComponent|1014|2016/01/01|COST_OF_LIVING|300|Y
MERGE|SalaryComponent|1014|2016/01/01|ADJUSTMENT|100|Y
```

Deleting a Salary

Use HCM Data Loader to delete salary records only if many records were created or loaded incorrectly and must be deleted in bulk. Unless you're deleting all salary records for an assignment, only the salary record with the latest salary start date can be deleted. For example, if a salary record has salary start dates of January 1, March 1, and September 1 then you can delete just the September 1 record. Alternatively, you can delete all three records.

This example Salary.dat file deletes an existing salary record. The Salary object is identified by its source key.

```
METADATA|Salary|SourceSystemOwner|SourceSystemId|AssignmentId(SourceSystemId)|DateFrom
DELETE|Salary|VISION|1012SAL16|EJ1012_EMP_ASG|2016/01/01
```

This example Salary.dat file deletes an existing salary. The Salary object is identified by its user key.

```
METADATA|Salary|AssignmentNumber|DateFrom
DELETE|Salary|1014|2016/01/01
```

Loading Salary Range Differential Profiles and Values: Examples

Salary range differential values are multipliers applied to grade rates to adjust them on the basis of worker location. For example, a differential value of 1.2 could be applied in one location while a differential value of 0.80 could be applied in another. These values vary to reflect the cost of living in these locations or the relative ease of hiring, for example. Salary range differential values are associated with a salary range differential profile, which you link to the salary basis. This topic provides examples showing how to load differential profiles and their differential values using HCM Data Loader.

Creating Salary Range Differential Profiles and Values

To create a differential profile, you load the Salary Range Differential object and one or more Salary Range Differential Value components. Each Salary Range Differential Value component is for a single location. Differential profiles reference legislative data groups, and their differential values reference locations. Referenced legislative data groups and locations must exist in the target environment before you load Salary Range Differential objects.

This example SalaryRangeDifferential.dat file creates a differential profile with a single differential value. Both components are identified using source keys.

```
METADATA|SalaryRangeDifferential|EffectiveEndDate|EffectiveStartDate|Name|Code|Status|Criteria|SourceSystemId|
SourceSystemOwner|LegislativeDataGroupName
MERGE|SalaryRangeDifferential||2000/01/01|SalRangeDiffName333|SalRangeDiffCode333|ACTIVE|LOCATION|
SalRangeDiffSSID_333|VISION|PM DE LDG
METADATA|SalaryRangeDifferentialValue|EffectiveEndDate|EffectiveStartDate|SourceSystemId|SourceSystemOwner|
RangeDiffId(SourceSystemId)|LocationCode|LocationSetCode|Differential
MERGE|SalaryRangeDifferentialValue||2000/01/01|SalRangeDiffVal_SSID_333|VISION|SalRangeDiffSSID_333|Atlanta|
Vision Corporation 202|0.2
```

This example SalaryRangeDifferential.dat file creates a differential profile with a single differential value. Both components are identified using user keys.

```
METADATA|SalaryRangeDifferential|EffectiveEndDate|EffectiveStartDate|Name|Code|Status|Criteria|
LegislativeDataGroupName
MERGE|SalaryRangeDifferential||2000/01/01|SalRangeDiffName444|SalRangeDiffCode444|ACTIVE|LOCATION|PM DE LDG
METADATA|SalaryRangeDifferentialValue|EffectiveEndDate|EffectiveStartDate|RangeDiffCode|LocationCode|
LocationSetCode|Differential
MERGE|SalaryRangeDifferentialValue||2000/01/01|SalRangeDiffCode444|Atlanta|Vision Corporation 202|0.2
```

Making Date-Effective Changes to Salary Range Differentials: Explained

The Salary Range Differential object is date-effective. That is, a history of changes to the Salary Range Differential object and its child components is maintained and can be viewed. For example, users can see the date-effective history of a differential

profile in the Compensation work area. This topic describes how date-effective records are managed for Salary Range Differential objects when you use HCM Data Loader.

Creating Date-Effective Records

When you create or update a differential value, a date-effective record is added to the differential value. A date-effective record must also be added to the parent differential profile if it has no date-effective record for that effective start date. In this case, you must include the parent Salary Range Differential component in the .dat file. It must have the same effective start date as the child differential value.


This table shows a history of changes to a differential profile.

RangeDiffId	EffectiveStartDate	EffectiveEndDate	Status	Criteria	LegislativeDataGrp	Code	Name
123	01-01-1951	01-05-2016	Active	Location	US1	US SRD	US Range
123	02-05-2016	31-12-4712	Active	Location	US1	US SRD	US Range

This table shows the related changes to its differential values.

RangeDiffValueId	EffectiveStartDate	EffectiveEndDate	Differential	LocationId
234	01-01-1951	31-12-4712	1.3	A
345	02-05-2016	31-12-4712	0.8	B
456	02-05-2016	31-12-4712	0.9	C

When the location B value was created, a date-effective record with the location B effective start date was also added to the differential profile. However, when the location C value was created, no date-effective record was added to the differential profile. A record for that effective start date already existed in the differential profile, so no additional record was needed. If you create or update multiple differential values on the same date, then only one date-effective record is added to the parent differential profile.

 **Note:** You don't have to include the parent Salary Range Differential component in the .dat file if it already has a date-effective record with the required effective start date.

Editing the Salary Range Differential Component

If you correct or update the parent Salary Range Differential object, then no date-effective record is added to its child components. For example, if you make the differential profile inactive, then a new date-effective record is added to the parent component. However, no date-effective record is added to the differential values.

12 Loading Work Structures

Loading Work Structures: Overview

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

Reviewing Component Hierarchies

Select a work structure on the Initiate Data Load page to review its component hierarchy.

Finding Example .dat Files

For examples of .dat files showing how to load work structures, see HCM Data Loader: Loading Work Structures (2021212.1) on My Oracle Support at <https://support.oracle.com>.

Loading Set-Enabled Work Structures

These work structures are set enabled:

- Departments
- Grades
- Jobs
- Locations

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

Using Action Reasons

You can associate action reasons with these work structures:

- Grades
- Grade rates
- Grade ladders
- Jobs
- Locations

- Organizations
- Positions

If you plan to use action reasons, then you must create them before you load work structures. Use the Manage Action Reasons task in the Setup and Maintenance work area.

Related Topics

- [Finding Business-Object Information: Explained](#)

Loading Grades: Explained

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

Grade Steps

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

Changing Logical Start and End Dates for Grades

You can't change either the first effective start date or last effective end date for an existing grade. Create grade records with effective start dates on or before the start dates of other objects that refer to your grades.

Loading Translated Grade and Grade-Step Names

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

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

```
METADATA|GradeTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Language|GradeName
MERGE|GradeTranslation|VISION|ADMIN|1951/01/01|FR|Administrateur
MERGE|GradeTranslation|VISION|IC1|1951/01/01|FR|Contributeur individuel
MERGE|GradeTranslation|VISION|M1|1951/01/01|FR|Superviseur
MERGE|GradeTranslation|VISION|D1|1951/01/01|FR|Directeur
```

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

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

Deleting Grades

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

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

Loading Grade Ladders: Explained

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

The Grade-Ladder Component Hierarchy

The grade ladder is made up of these components:

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

Loading Translated Grade-Ladder and Step-Rate Names

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

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

Deleting Grade Ladders

You can delete a grade ladder using HCM Data Loader, provided that it's not referred to by another object, such as an assignment. Deleting a grade ladder doesn't delete the associated grades. However, if you delete a grade ladder with steps, then the associated step rates and step-rate values are deleted automatically. This example GradeLadder.dat file deletes an existing grade ladder. It identifies the grade ladder by its source key.

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

Loading Grade Rates: Explained

A grade rate is a collection of general information about a payment type, such as salary, bonus, or overtime payments. It includes details such as the currency, frequency of the payment, and annualization factor. For example, the Overtime Rate Analysts grade rate could provide this information for the overtime payments made to analysts. Grade rates are associated with grade-rate values. Each associated grade-rate value identifies a grade and either a fixed payment or a payment range for the grade. For a payment range, you must supply both the minimum and maximum values. This topic describes aspects of the grade-rate object that you must understand to load grade rates successfully.

Loading Referenced Objects

Grade rates and grade-rate values are created for a legislative data group. Therefore, you must load legislative data groups before loading grade rates.

Loading Translated Grade-Rate Names

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

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

Deleting Grade Rates

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

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

Loading Jobs: Explained

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

Loading Referenced Objects

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

Benchmark Jobs

To use benchmark jobs:

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

Loading Translated Job Names

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

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

Deleting Jobs

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

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

Loading Job Families: Examples

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

Creating Job Families Using Source Keys

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

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

Loading Translated Job-Family Names

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

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

Deleting Job Families

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

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

Loading Locations: Explained

A location is the physical address of a place where you conduct business or that's of interest to your business. The location object records the address and other details, such as the location name, description, phones, and billing and shipping information. This topic describes aspects of the location object that you must understand to load locations successfully.

Changing Logical Start and End Dates for Locations

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

Loading Translated Location Names

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

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

Deleting Locations

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


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

Loading Organizations: Explained

An organization can be any organizational unit, such as a department or division, in your enterprise. Organizations provide the framework for performing legal reporting, financial control, and management reporting. This topic describes aspects of the organization object that you must understand to load organizations successfully.

Organization Components


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


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

Loading Referenced Objects

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

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

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

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

Organization Classification Codes

An organization can have one or more classifications. You can supply either the code or its name. Valid classifications are:

Classification Name	Classification Code
Bargaining Association	HCM_BARGAINING_ASSOCIATION
Department	DEPARTMENT
Disability Organization	HCM_DISABILITY_ORGANIZATION
Division	HCM_DIVISION
Enterprise	ENTERPRISE
Legal Employer	HCM_LEMP

Classification Name	Classification Code
Legal Reporting Unit	HCM_LRU
Payroll Statutory Unit	HCM_PSU
Reporting Establishment	HCM_REPORTING_ESTABLISHMENT

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

Loading Translated Organization Names

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

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


Deleting Organizations

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

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

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

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

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

Loading Positions: Explained

A position is a single occurrence of a job in a specified department. Positions may also be restricted by location. For example, you could create the HR Manager Sales France position for the HR Manager job in the Sales department in France. This topic describes aspects of the position object that you must understand to load positions successfully.

Effective Start Dates

A position and its child components must have the same effective start date.

Loading Referenced Objects

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

- Positions exist in business units. Ensure that referenced business units exist.
- Both **Job** and **Department** are required attributes of a position. Ensure that referenced jobs and departments exist.
- You can associate a location and valid grades with a position. Ensure that any referenced locations and grades exist.

Position Type

When the position type is **Single incumbent**, you must supply:

- An FTE value. The value must be less than or equal to 1. For example, if the position incumbent works half the standard working hours, then the full-time equivalent value is 0.5.
- A headcount of 1.

When the position type is **Shared**, you must supply:

- An FTE value
- A headcount that's equal to or greater than 1

Loading Translated Position Names

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

```
METADATA|PositionTranslation|SourceSystemOwner|SourceSystemId|EffectiveStartDate|Language|Name
MERGE|PositionTranslation|VISION|APP_SALES|2015/01/01|FR|Sales Consultant d'application
```

Deleting Positions

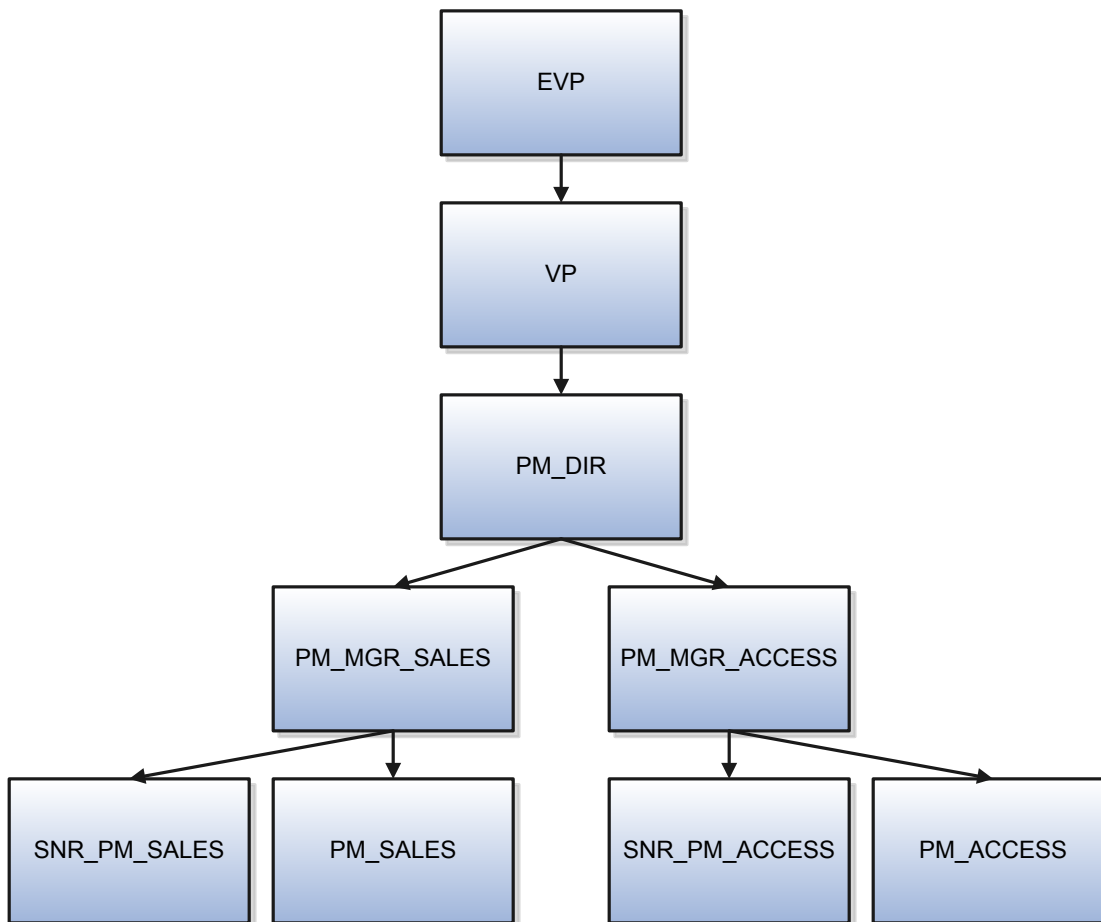
You can delete a position using HCM Data Loader, provided that it has never had an incumbent. When you delete a position, its child components are deleted automatically. This example Position.dat file deletes an existing position. The position is identified by its source key.

```
METADATA|Position|SourceSystemOwner|SourceSystemId|EffectiveStartDate
DELETE|Position|VISION|APP_SALES|2015/01/01
```

Loading HCM Position Hierarchies: Examples

An HCM position hierarchy is derived from the relationships between positions and their parent positions. You can identify a position's parent position on the Manage Positions page if the **Use HCM Position Hierarchy** option is enabled for the enterprise. This topic provides examples of how to load HCM position hierarchy objects using HCM Data Loader. Each position hierarchy object identifies a position and its parent. All positions must exist in the target environment.

This figure shows the position hierarchy that these examples load.



Loading HCM Position Hierarchies Using Source Keys

This example PositionHierarchy.dat file loads HCM position hierarchy objects using source keys.

```

METADATA|PositionHierarchy|SourceSystemOwner|SourceSystemId|EffectiveStartDate|EffectiveEndDate|
ParentPositionId(SourceSystemId)|PositionId(SourceSystemId)
MERGE|PositionHierarchy|Vision|1|2015/01/01|4712/12/31|PM_MGR_ACCESS|PM_ACCESS
MERGE|PositionHierarchy|Vision|2|2015/01/01|4712/12/31|PM_MGR_ACCESS|SNR_PM_ACCESS
MERGE|PositionHierarchy|Vision|3|2015/01/01|4712/12/31|PM_MGR_SALES|PM_SALES
MERGE|PositionHierarchy|Vision|4|2015/01/01|4712/12/31|PM_MGR_SALES|SNR_PM_SALES
MERGE|PositionHierarchy|Vision|5|2015/01/01|4712/12/31|PM_DIR|PM_MGR_ACCESS
MERGE|PositionHierarchy|Vision|6|2015/01/01|4712/12/31|PM_DIR|PM_MGR_SALES
MERGE|PositionHierarchy|Vision|7|2015/01/01|4712/12/31|VP|PM_DIR
MERGE|PositionHierarchy|Vision|8|2015/01/01|4712/12/31|EVP|VP
  
```

Loading HCM Position Hierarchies Using User Keys

This example PositionHierarchy.dat file loads HCM position hierarchy objects using user keys.

```
METADATA|PositionHierarchy|EffectiveStartDate|EffectiveEndDate|ParentBusinessUnitName|ParentPositionCode|  
BusinessUnitName|PositionCode  
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_MGR_ACCESS|VisionBU|PM_ACCESS  
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_MGR_ACCESS|VisionBU|SNR_PM_ACCESS  
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_MGR_SALES|VisionBU|PM_SALES  
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_MGR_SALES|VisionBU|SNR_PM_SALES  
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_DIR|VisionBU|PM_MGR_ACCESS  
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|PM_DIR|VisionBU|PM_MGR_SALES  
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|VP|VisionBU|PM_DIR  
MERGE|PositionHierarchy|2015/01/01|4712/12/31|VisionBU|EVP|VisionBU|VP
```

Running the Flatten HCM Position Hierarchy Process

Changes to the parent position on the Manage Positions page automatically trigger a process to update the position hierarchy. However, this process isn't triggered when you load position hierarchies using HCM Data Loader. You must run the Flatten HCM Position Hierarchy process explicitly after loading position hierarchy objects using HCM Data Loader.

13 Loading Hierarchy Trees

Loading Department Trees: Explained

A department tree is a hierarchy of departments. For each tree, you can define multiple versions. However, only one version can be active on any date. This topic describes aspects of department-tree objects that you must understand to load them successfully using HCM Data Loader.

Key Support

Department tree is unlike most other objects that you can load using HCM Data Loader, as it doesn't support the integration-key types. When you load department trees, you must supply the user-key attributes.

How Department Trees Are Implemented

Department trees are implemented using two HCM Data Loader objects. This table introduces those objects.

Object	Description
Department tree	The definition of a tree and the versions of the tree. A tree version is empty until you load department tree nodes for the tree version.
Department tree node	The department tree nodes for a tree version. Each node represents a department.

Loading Department Trees

You supply department-tree data in the DepartmentTree.dat file for processing by HCM Data Loader. The **TreeStructureCode** attribute of both the department tree and the department tree version must be set to **PER_DEPT_TREE_STRUCTURE** for a department tree.


 **Note:** When you load a new department tree version for an existing department tree, you must include the department tree component in the DepartmentTree.dat file. You can't load a new department tree version in isolation.

This example DepartmentTree.dat file creates a department tree with code **Department Tree Code** and three versions, one for each of January, February, and March.

```
METADATA|DepartmentTree|TreeStructureCode|TreeCode|TreeName|TreeDescription
MERGE|DepartmentTree|PER_DEPT_TREE_STRUCTURE|Department Tree Code|My Tree Name|My Tree Description
METADATA|DepartmentTreeVersion|TreeStructureCode|TreeCode|TreeVersionName|TreeVersionDescription|
TreeVersionStartDate|TreeVersionEndDate
MERGE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan| Tree Description
Jan|2015/01/01|2015/01/31
MERGE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Feb| Tree Description
Feb|2015/02/01|2015/02/28
MERGE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Mar| Tree Description
Mar|2015/03/01|2015/03/31
```

Deleting Department Trees

You can delete department trees and department tree versions using HCM Data Loader. When you delete a department tree, the tree and all of its versions and nodes are deleted.

 **Note:** The departments themselves aren't deleted. Only the relationships defined in the tree structure are deleted.

This example DepartmentTree.dat file deletes a department tree:

```
METADATA|DepartmentTree|TreeStructureCode|TreeCode
DELETE|DepartmentTree|PER_DEPT_TREE_STRUCTURE|Department Tree Code
```

If you delete a department tree version, then only the specified version and its nodes are deleted. The tree and its other versions are retained. This example DepartmentTree.dat file deletes a department tree version:

```
METADATA|DepartmentTreeVersion|TreeStructureCode|TreeCode|TreeVersionName
DELETE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Feb
```

Loading Department Tree Nodes: Explained

A department is an internal organization to which workers are assigned. You can add any organization that has the Department classification as a tree node to a department tree version. This topic describes aspects of department tree nodes that you must understand to load them successfully using HCM Data Loader.

Preparing to Load Department Tree Nodes

Before you load department tree nodes:

- You must create the department tree and department tree version to which the nodes belong. You can either load them using HCM Data Loader or create them using the Manage Department Trees task in the Workforce Structures work area.
- The departments that the tree nodes refer to must exist on the start date of the tree hierarchy version. Simply adding a department to the department tree hierarchy doesn't create the department. Similarly, removing a department reference from a department tree hierarchy doesn't delete the department record.

Loading Department Tree Nodes

The department tree node records the relationship between a parent department and a child, which can be either a department or another department tree version. When the child is another department tree version, the department nodes of that version are included by reference. When you load department tree nodes, these rules apply:

- You must supply a parent department for all nodes except the top-level node of your department tree version. The parent departments must exist.
- The **TreeStructureCode** attribute must be set to **PER_DEPT_TREE_STRUCTURE**.
- To avoid circular relationships, a department can appear only once in each tree version.
- Each tree version can have only one top node. If you need multiple peer departments at the top, then consider using a dummy department as the top node.

Related Topics

- Processes to Run After Loading Data: Explained

Loading Department Tree Nodes: Examples

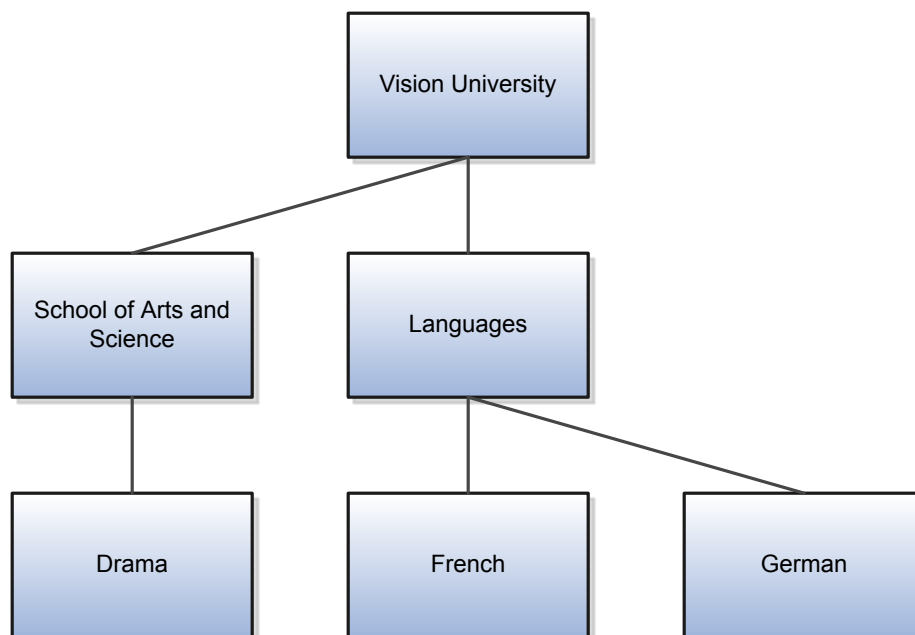
You can load any organization that has the Department classification as a tree node to a department tree version. This topic shows how to load department tree nodes using HCM Data Loader.

Creating Department Tree Nodes

This example DepartmentTreeNode.dat file loads department tree nodes to an existing department tree version.

```
METADATA|DepartmentTreeNode|TreeStructureCode|TreeCode|TreeVersionName|DepartmentName|ParentDepartmentName
MERGE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|Vision University|
MERGE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|School of Arts and
Science|Vision University
MERGE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|Drama|School of Arts
and Science
MERGE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|Languages|Vision
University
MERGE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|French|Languages
MERGE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|German|Languages
```

This figure shows the hierarchy nodes that this DepartmentTreeNode.dat file creates.

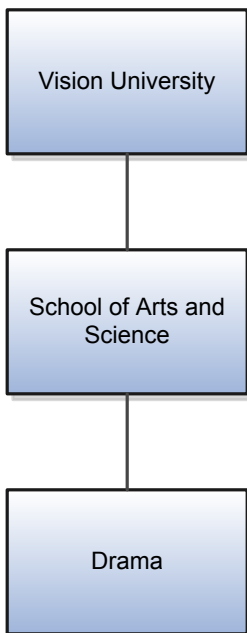


Deleting Department Tree Nodes

You can delete department tree nodes using HCM Data Loader. The departments are removed from the tree version, but the department objects themselves continue to exist. To delete a department tree node and all nodes below it in the department tree hierarchy, set the **DeleteChildNodesFlag** attribute to **Y**. This example DepartmentTreeNode.dat file deletes a node and its child nodes.

```
METADATA|DepartmentTreeNode|TreeStructureCode|TreeCode|TreeVersionName|DepartmentName|DeleteChildNodesFlag
DELETE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|Languages|Y
```

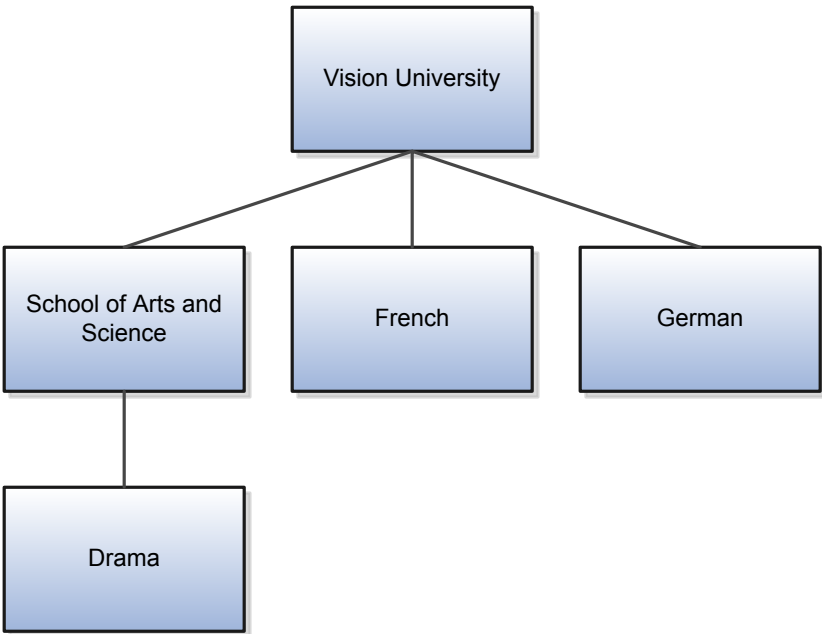
This figure shows the hierarchy after deleting the Languages node and its child nodes.



To delete a department tree node and promote its child nodes, set the **DeleteChildNodesFlag** attribute to **N**. This example DepartmentTreeNode.dat file deletes a node and promotes its child nodes.

```
METADATA|DepartmentTreeNode|TreeStructureCode|TreeCode|TreeVersionName|DepartmentName|DeleteChildNodesFlag
DELETE|DepartmentTreeVersion|PER_DEPT_TREE_STRUCTURE|Department Tree Code|Tree Version Jan|Languages|N
```

This figure shows the hierarchy after deleting the Languages node and promoting its child nodes.



Loading Organization Trees: Explained

An organization tree is a hierarchy of organizations of any classification. For each tree, you can define multiple versions. However, only one version can be active on any date. This topic describes aspects of organization-tree objects that you must understand to load them successfully using HCM Data Loader.

Key Support

Organization tree is unlike most other objects that you can load using HCM Data Loader, as it doesn't support the integration-key types. When you load organization trees, you must supply the user-key attributes.


How Organization Trees Are Implemented

Organization trees are implemented using two HCM Data Loader objects. This table introduces those objects.

Object	Description
Organization tree	The definition of a tree and the versions of the tree. A tree version is empty until you load organization tree nodes for the tree version.
Organization tree node	The organization tree nodes for a tree version. Each node represents an organization.

Loading Organization Trees

You supply organization-tree data in the OrganizationTree.dat file for processing by HCM Data Loader. The **TreeStructureCode** attribute of both the organization tree and the organization tree version must be set to **PER_ORG_TREE_STRUCTURE** for an organization tree.


 **Note:** When you load a new organization tree version for an existing organization tree, you must include the organization tree component in the OrganizationTree.dat file. You can't load a new organization tree version in isolation.

This example OrganizationTree.dat file creates an organization tree with code Organization Tree Code and three versions, one for each of January, February, and March.

```
METADATA|OrganizationTree|TreeStructureCode|TreeCode|TreeName|TreeDescription
MERGE|OrganizationTree|PER_ORG_TREE_STRUCTURE|Organization Tree Code|My Tree Name|My Tree Description
METADATA|OrganizationTreeVersion|TreeStructureCode|TreeCode|TreeVersionName|TreeVersionDescription|
TreeVersionStartDate|TreeVersionEndDate
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Organization Tree Code|Tree Version Jan| Tree Description
Jan|2015/01/01|2015/01/31
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Organization Tree Code|Tree Version Feb| Tree Description
Feb|2015/02/01|2015/02/28
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Organization Tree Code|Tree Version Mar| Tree Description
Mar|2015/03/01|2015/03/31
```

Deleting Organization Trees

You can delete organization trees and organization tree versions using HCM Data Loader. When you delete an organization tree, the tree and all of its versions and nodes are deleted.

 **Note:** The organizations themselves aren't deleted. Only the relationships defined in the tree structure are deleted.

This example OrganizationTree.dat file deletes an organization tree:

```
METADATA|OrganizationTree|TreeStructureCode|TreeCode
DELETE|OrganizationTree|PER_ORG_TREE_STRUCTURE|Organization Tree Code
```

If you delete an organization tree version, then only the specified version and its nodes are deleted. The tree and its other versions are retained. This example OrganizationTree.dat file deletes an organization tree version:

```
METADATA|OrganizationTreeVersion|TreeStructureCode|TreeCode|TreeVersionName
DELETE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Organization Tree Code|Tree Version Feb
```

Loading Organization Tree Nodes: Explained

You can load an organization of any classification as a tree node to an organization tree version. This topic describes aspects of organization tree nodes that you must understand to load them successfully using HCM Data Loader.

Preparing to Load Organization Tree Nodes

Before you load organization tree nodes:

- You must create the organization tree and organization tree version to which the nodes belong. You can either load them using HCM Data Loader or create them using the Manage Organization Trees task in the Workforce Structures work area.
- The organizations that the tree nodes refer to must exist on the start date of the tree hierarchy version. Simply adding an organization to the organization tree hierarchy doesn't create the organization. Similarly, removing an organization reference from an organization tree hierarchy doesn't delete the organization record.

Loading Organization Tree Nodes

The organization tree node records the relationship between a parent organization and a child, which can be either an organization or another organization tree version. When the child is an organization, you specify its classification on the **ClassificationCode** attribute. When the child is another organization tree version, the organization nodes of that version are included by reference. When you load organization tree nodes, these rules apply:

- You must supply a parent organization for all nodes except the top-level node of your organization tree version. The parent organizations must exist.
- The **TreeStructureCode** attribute must be set to **PER_ORG_TREE_STRUCTURE**.
- To avoid circular relationships, an organization can appear only once in each tree version.
- Each tree version can have only one top node. If you need multiple peer organizations at the top, then consider using the enterprise as the top node.

Related Topics

- [Processes to Run After Loading Data: Explained](#)

Loading Organization Tree Nodes: Examples

You can load an organization with any classification as a tree node to an organization tree version. This topic shows how to load organization tree nodes using HCM Data Loader.

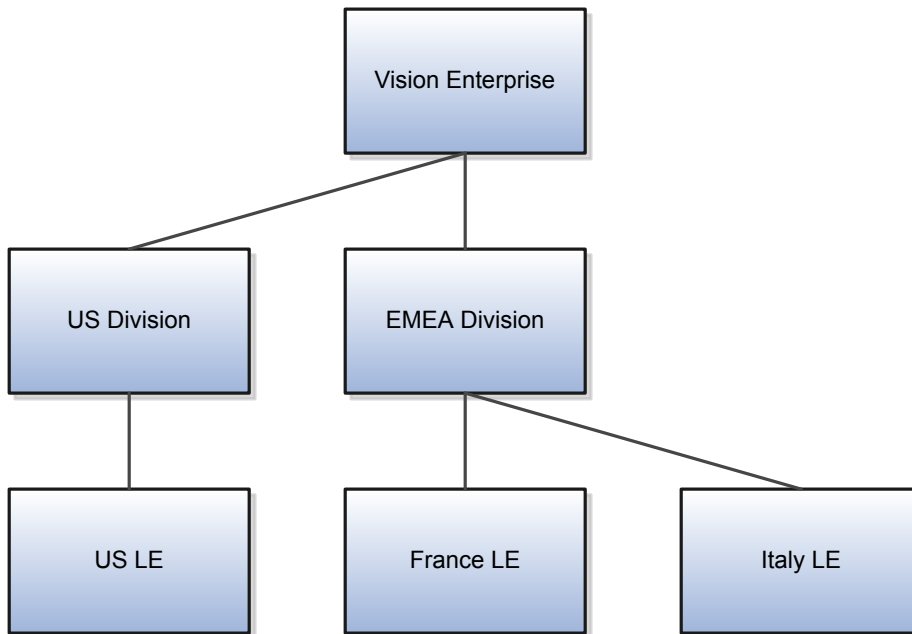
Creating Organization Tree Nodes

This example OrganizationTreeNode.dat file loads organization tree nodes to an existing organization tree version.

```
METADATA|OrganizationTreeNode|TreeStructureCode|TreeCode|TreeVersionName|OrganizationName|ClassificationCode|
ParentOrganizationName|ParentOrganizationClassificationCode
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|Vision Enterprise|
Enterprise||
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|US Division|Division|
Vision Enterprise|Enterprise
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|US LE|Legal Employer|
US Division|Division
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|EMEA Division|
Division|Vision Enterprise|Enterprise
```

```
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|France LE|Legal
Employer|EMEA Division|Division
MERGE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|Italy LE|Legal
Employer|EMEA Division|Division
```

This figure shows the hierarchy nodes that this OrganizationTreeNode.dat file creates.

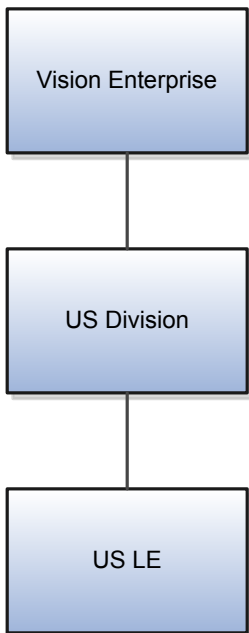


Deleting Organization Tree Nodes

You can delete organization tree nodes using HCM Data Loader. The organizations are removed from the tree version, but the organization objects themselves continue to exist. To delete an organization tree node and all nodes below it in the organization tree hierarchy, set the **DeleteChildNodesFlag** attribute to **Y**. This example OrganizationTreeNode.dat file deletes a node and its child nodes.

```
METADATA|OrganizationTreeNode|TreeStructureCode|TreeCode|TreeVersionName|OrganizationName|ClassificationCode|
DeleteChildNodesFlag
DELETE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|EMEA Division|
Division|Y
```

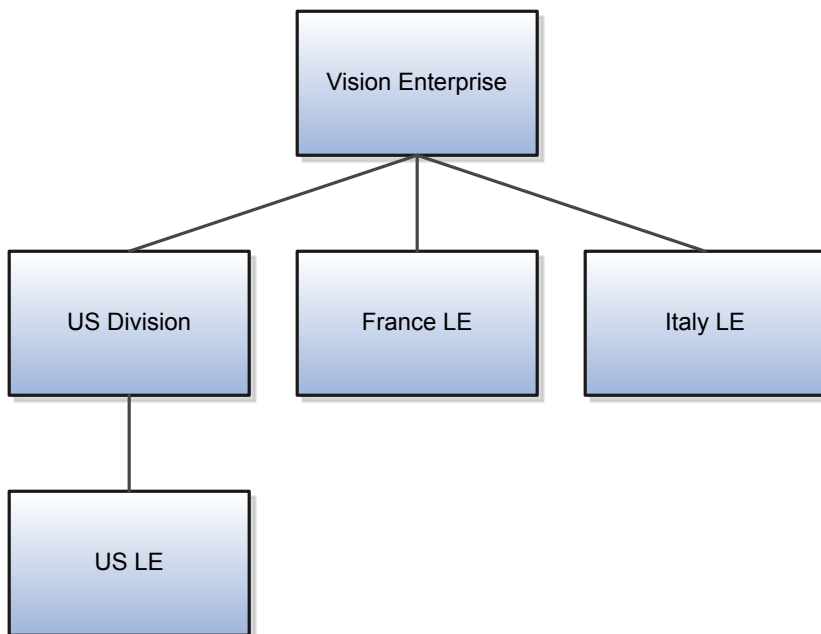
This figure shows the hierarchy after deleting the EMEA Division node and its child nodes.



To delete an organization tree node and promote its child nodes, set the **DeleteChildNodesFlag** attribute to **N**. This example OrganizationTreeNode.dat file deletes a node and promotes its child nodes.

```
METADATA|OrganizationTreeNode|TreeStructureCode|TreeCode|TreeVersionName|OrganizationName|ClassificationCode|
DeleteChildNodesFlag
DELETE|OrganizationTreeVersion|PER_ORG_TREE_STRUCTURE|Vision Tree Code|Tree Version 2017|EMEA Division|
Division|N
```

This figure shows the hierarchy after deleting the EMEA Division node and promoting its child nodes.



14 Loading Talent Objects

Loading Goals, Goal Plans, and Goal Plan Sets: Overview

You can load goals, goal plans, and goal plan sets using HCM Data Loader. Most of these objects have dependencies on other objects. This table identifies the other objects that must exist in the target environment. If you're loading some or all of those objects with goals, goal plans, or goal plan sets, then follow the load order shown here.

Object	Object Load Order	Notes
Goal plan	<ol style="list-style-type: none"> 1. Worker and primary assignment 2. Eligibility profile 3. Organization 4. Goal plan document type 5. Goal plan 	<ul style="list-style-type: none"> Review periods must exist in the target environment before you upload associated goal plans. Organizations are required for organization goal plans. A goal plan document type component is required when the IncludeInPerfDoc attribute is set to Y.
Goal plan set	<ol style="list-style-type: none"> 1. Goal plan 2. Worker and primary assignment 3. Eligibility profile 4. Goal plan set 	<ul style="list-style-type: none"> Review periods must exist in the target environment before you upload associated goal plan sets. Goal plan set objects can refer to goal plans.
Library goal	<ol style="list-style-type: none"> 1. Content item 	Library goals are loaded as content items.
Organization goal	<ol style="list-style-type: none"> 1. Worker (line manager) 2. Organization 3. Goal plan 4. Goal 	<p>For organization goals:</p> <ul style="list-style-type: none"> The line manager is the organization owner. Goal plans are required.
Template goal	<ol style="list-style-type: none"> 1. Worker (human resource specialist) and assignment 2. Goal plan 3. Goal 	Goal plans are required.
Worker development goal	<ol style="list-style-type: none"> 1. Worker 2. Content item 3. Goal 	Worker development goals aren't associated with goal plans.
Worker performance goal	<ol style="list-style-type: none"> 1. Worker and assignment 2. Goal plan 3. Content item 4. Goal 	Worker performance goals must be associated with goal plans.

Related Topics

- Goal Management: Explained

Loading Goals: Explained

Goals represent the objectives of employees and organizations. This topic describes aspects of the goal object that you must understand to load goals successfully.

Goal Types

This table identifies the types of goals that you can load using HCM Data Loader.

Goal Type	Description
Organization goal	Created by a manager for a specific organization.
Template goal	Used in goal plans.
Worker goal	Assigned to a worker by a manager or human resource specialist. Two types of worker goals exist: <ul style="list-style-type: none">• Development goals• Performance goals


 **Note:** You can also load library goals using HCM Data Loader. You load library goals as profile content items. You can refer to library goals from other goal objects.

When you load goals in the Goal.dat file, you can include the **GoalTypeCode** attribute. This value is validated using the HRG_GOAL_TYPE lookup type. For organization goals, **GoalTypeCode** must be PERFORMANCE, which is the default value.

Goal Components

The components that you can supply when loading goals depend on the goal type. This table identifies when you can provide each component.

Goal Component	Goal Type
Goal	All
Goal action	All
Goal plan goal	<ul style="list-style-type: none">• Worker performance goals• Organization goals• Template goals
Goal target outcome	<ul style="list-style-type: none">• Worker goals• Template goals
Goal measurement	All

Goal Component	Goal Type
 Note: You can specify only one goal measurement component for a development goal.	
Goal access	<ul style="list-style-type: none">Worker goals
Goal alignment	<ul style="list-style-type: none">Worker performance goalsOrganization goals

User Keys

You can't supply a user key as the unique identifier for a goal or its child components when you load them using HCM Data Loader. You must supply source keys when loading goals.

Profile Options

This table identifies profile options that must be both enabled and set to **Y** for successful loading of goal components.

Profile Option	What the Option Enables
HRG_ENABLE_GOAL_ALIGN	Alignment of goals in the goal alignment component
HRG_ENABLE_GRANT_ACCESS	Shared goals in the goal access component
HRG_ENABLE_OUTCOMES	Creation of goal target outcomes in the goal target outcome component
HRG_ENABLE_TASK	Creation of goal tasks in the goal action component

- To enable profile options, use the Manage Profile Options task.
- To set the value of a profile option, use the Manage Administrator Profile Values task.

Perform both tasks in the Setup and Maintenance work area.

Deleting Goals

You can delete goals using HCM Data Loader. When you delete a parent goal object, these child components are also deleted automatically:

- Goal access
- Goal action
- Goal alignment, where **GoalId** or **AlignedGoalId** identifies the goal that you're deleting
- Goal plan goal
- Goal target outcome

You can also delete child components individually, if required.

Goal measurement components aren't deleted automatically when you delete the parent goal using HCM Data Loader. You must delete goal measurement components explicitly before deleting the parent goal. This example Goal.dat file deletes a goal measurement component. The parent goal is identified by its source key.

```
METADATA|GoalMeasurement|SourceSystemId|SourceSystemOwner|GoalId(SourceSystemId)|
DELETE|GoalMeasurement|GoalMeasurement_001_SSID|VISION|WorkerGoal_001_SSID
```

This example Goal.dat file deletes a goal and its child components, other than goal measurement. The goal is identified by its source key.

```
METADATA|Goal|SourceSystemId|SourceSystemOwner|
DELETE|Goal|WorkerGoal_001_SSID|VISION
```


Loading Goals: Examples

This topic provides examples showing how to load worker goals and organization goals using HCM Data Loader.

Creating a Worker Goal

This example Goal.dat file creates an active performance goal for a worker. In this file:

- The user-key attributes **WorkerPersonNumber** and **WorkerAssignmentNumber** identify the worker and the relevant assignment.
- The user-key attribute **AssignedByPersonNumber** identifies the human resource specialist or line manager who's creating the goal.
- A goal plan goal component must be provided.

 **Note:** For development goals, a goal plan goal component isn't required.

```
METADATA|Goal|SourceSystemId|SourceSystemOwner|GoalName|Description|StartDate|TargetCompletionDate|
WorkerPersonNumber|WorkerAssignmentNumber|StatusCode|GoalSourceCode|AllowKeyAttrUpdateFlag|
IncludeInPerfdocFlag|AssignedByPersonNumber|
MERGE|Goal|WorkerGoal_001_SSID|VISION|WorkerGoal_001|WorkerGoal_001_Desc|2015/01/01|2015/12/31|8153787|
EEEE8153787|Not started|HR|Y|Y|8153756|
METADATA|GoalPlanGoal|SourceSystemId|SourceSystemOwner|GoalId(SourceSystemId)|GoalPlanName|GoalPlanStartDate|
GoalPlanEndDate|PriorityCode|
MERGE|GoalPlanGoal|GoalPlanGoal_001_SSID|VISION|WorkerGoal_001_SSID|GP_009_01|2015/01/01|2015/12/31|Medium
```

Creating a Worker Goal Against a Library Goal

This example Goal.dat file creates an active performance goal against a library goal for a worker. HCM Data Loader doesn't take values from the library goal. Instead, you must supply these values when you load the worker goal. The **ReferenceContentItemName** attribute is the user key for the library goal, which must exist.

```
METADATA|Goal|SourceSystemId|SourceSystemOwner|GoalName|Description|StartDate|TargetCompletionDate|
WorkerPersonNumber|WorkerAssignmentNumber|StatusCode|GoalSourceCode|AllowKeyAttrUpdateFlag|
IncludeInPerfdocFlag|AssignedByPersonNumber|ReferenceContentItemName|
MERGE|Goal|WorkerGoal_002_SSID|VISION|WorkerGoal_002|WorkerGoal_002_Desc|2015/01/01|2015/12/31|8153787|
EEEE8153787|Not started|HR|Y|Y|8153756|ZHRG-Completing Learning Path
```

Creating an Organization Goal

This example Goal.dat file creates and publishes an organization goal for a line manager. In this file:

- The user-key attribute **AssignedByPersonNumber** identifies the line manager who owns the organization goal. To publish the organization goal to the organization, set the **PublishedFlag** attribute to **Y**.
- The **GoalSourceCode** attribute is set to **Organization owner**. This value is required for organization goals.
- A goal plan goal component must be provided.

```
METADATA|Goal|SourceSystemId|SourceSystemOwner|GoalName|Description|StartDate|TargetCompletionDate|
OrganizationName|StatusCode|GoalSourceCode|AllowKeyAttrUpdateFlag|AssignedByPersonNumber|PublishedFlag
MERGE|Goal|OrganizationGoal_001_SSID|VISION|OrganizationGoal_001|OrganizationGoal_001_Desc|2015/01/01|
2015/12/31|Vision Corporation Enterprise|Not started|Organization owner|Y|8153786|Y
METADATA|GoalPlanGoal|SourceSystemId|SourceSystemOwner|GoalId(SourceSystemId)|GoalPlanName|GoalPlanStartDate|
GoalPlanEndDate|PriorityCode
MERGE|GoalPlanGoal|GoalPlanGoal_002_SSID|VISION|OrganizationGoal_001_SSID|GP_009_01|2015/01/01|2015/12/31|
Medium
```

Loading Goal Plans: Explained

Goal plans are used for the mass assignment of goals to worker populations. This topic describes aspects of the goal plan object that you must understand to load goal plans successfully.

Assigning Goals to Workers

When you load a goal plan object, it's not automatically assigned to workers. You can assign the goal plan to workers in either of these ways:

- Schedule goal plan assignments using the Manage Goal Plans task.
- Load the goal plan assignment component of the goal plan.

The Mass Request Component

The mass request component identifies who's requesting the goal plan. You can identify the plan requester on the goal plan itself by supplying the **RequestSubmissionDate** and **ReqSubmittedByPersonNumber** attributes, for example. In this case, the mass request component is created automatically and you don't have to supply it. However, you must supply a mass request component when you create a goal plan:

- If you don't identify the plan requester on the goal plan object.
- To associate a source key with the mass request record.

Alternatively, you can update the default source key later using the source key object.

- If you're loading child components of the mass request component.

The child components of the mass request component identify the employee population to which the goal plan is to be assigned.

Deleting Goal Plans

You can delete goal plans using HCM Data Loader. However, don't delete goal plans that have been assigned to workers. When you delete a parent goal plan object, these child components are also deleted automatically:

- Goal plan document type
- Mass request

 **Note:** When you delete a mass request component, its child components are deleted automatically.

You can also delete child components individually, if required.

Goal plan goal and goal plan assignment components aren't deleted automatically when you delete the parent goal plan using HCM Data Loader. You must delete those components explicitly before you delete the parent goal plan. This example GoalPlan.dat file deletes the goal plan goal record that links goal plan GP_001_01 with template goal TemplateGoal_001.

```
METADATA|GoalPlanGoal|SourceSystemId|SourceSystemOwner|GoalPlanExternalId
DELETE|GoalPlanGoal|GPG_001_01_SSID|VISION|GP_001_01
```

This example Goal.dat file deletes the template goal TemplateGoal_001 (SourceSystemId: TemplateGoal_001_SSID).

```
METADATA|Goal|SourceSystemId|SourceSystemOwner
DELETE|Goal|TemplateGoal_001_SSID|VISION
```

This example GoalPlan.dat file deletes the goal plan GP_001_01 and relevant child records.

```
METADATA|GoalPlan|GoalPlanExternalId
DELETE|GoalPlan|GP_001_01
```

Related Topics

- [Goal Plans: Explained](#)

Loading Goal Plans: Examples

This topic provides examples showing how to load goal plans using HCM Data Loader.

Creating a Goal Plan

This example GoalPlan.dat file loads a goal plan without a mass request component. In this case, a mass request record for the goal plan is created automatically using the values passed for the **ReqSubmittedByPersonNumber** and **RequestSubmissionDate** attributes.

```
METADATA|GoalPlan|GoalPlanExternalId|GoalPlanName|GoalPlanTypeCode|GoalPlanActiveFlag|EnableWeightingFlag|
StartDate|EndDate|EnforceGoalWeightFlag|GoalAccessLevelCode|IncludeInPerfdocFlag|ReqSubmittedByPersonNumber|
RequestSubmissionDate|ReviewPeriodName
MERGE|GoalPlan|GP_001_01|GP_001_01|Performance|A|Y|2015/01/01|2015/12/31|Y|HR specialist and manager|N|
8153756|2015/09/20|Default Review Period - Vision Corporation Enterprise
```

Creating a Goal Plan with Document Types

This example GoalPlan.dat file creates a goal plan with the **IncludeInPerfdocFlag** attribute set to **Y** and assigns a document type.

```
METADATA|GoalPlan|GoalPlanExternalId|GoalPlanName|GoalPlanTypeCode|GoalPlanActiveFlag|EnableWeightingFlag|
StartDate|EndDate|EnforceGoalWeightFlag|GoalAccessLevelCode|IncludeInPerfdocFlag|ReqSubmittedByPersonNumber|
RequestSubmissionDate|ReviewPeriodName
MERGE|GoalPlan|GP_001_02|GP_001_02|Performance|A|Y|2016/01/01|2016/12/31|Y|HR specialist and manager|Y|
8153756|2016/10/04|Default Review Period - Vision Corporation Enterprise
METADATA|GoalPlanDocTypes|GoalPlanExternalId|DocTypeName
MERGE|GoalPlanDocTypes|GP_001_02|Annual Evaluations
```

Creating a Goal Plan with Eligibility Criteria

This example GoalPlan.dat file creates a goal plan with mass request and eligibility profile object components.

```
METADATA|GoalPlan|GoalPlanExternalId|GoalPlanName|GoalPlanTypeCode|GoalPlanActiveFlag|EnableWeightingFlag|
StartDate|EndDate|EnforceGoalWeightFlag|GoalAccessLevelCode|IncludeInPerfdocFlag|ReviewPeriodName
MERGE|GoalPlan|GP_001_04|GP_001_04|Performance|A|Y|2016/01/01|2016/12/31|Y|HR specialist and manager|N|Default
Review Period - Vision Corporation Enterprise
METADATA|MassRequest|GoalPlanExternalId|ReqSubmittedByPersonNumber
MERGE|MassRequest|GP_001_04|8153756
METADATA|EligibilityProfileObject|GoalPlanExternalId|EligibilityProfileName|RequiredFlag
MERGE|EligibilityProfileObject|GP_001_04|Goals_EligibilityProfile_Software Developer 4|Y
```

Loading Goal Plan Sets: Explained

Goal plan sets are used for the mass assignment of related goal plans to worker populations. This topic describes aspects of the goal plan set object that you must understand to load goal plan sets successfully.

Assigning Goal Plan Sets to Workers

When you load a goal plan set, it's not automatically assigned to workers. You can assign the goal plan set to workers in either of these ways:

- Schedule goal plan set assignments using the Manage Goal Plan Sets task.
- Load the goal plan assignment component of the goal plan object. You may use this approach, for example, to assign goals following the transfer of workers from one legal entity to another.

The Mass Request Component

The mass request component identifies who's requesting the goal plan set. You can identify the plan requester on the goal plan set itself by supplying the **RequestSubmissionDate** and **ReqSubmittedByPersonNumber** attributes, for example. In this case, the mass request component is created automatically and you don't have to supply it. However, you must supply a mass request component when you create a goal plan set:

- If you don't identify the plan requester on the goal plan set object.

- To associate a source key with the mass request record.

Alternatively, you can update the default source key later using the source key object.

- If you're loading child components of the mass request component.

The child components of the mass request component identify the employee population to which the goal plan set is to be assigned.

Deleting Goal Plan Sets

You can delete goal plan sets using HCM Data Loader. When you delete a parent goal plan set, these child components are also deleted automatically:

- Goal plan set plan
- Mass request

 **Note:** When you delete a mass request component, its child components are deleted automatically.

Don't delete goal plan sets that have already been assigned to workers.

This example GoalPlanSet.dat file deletes a goal plan set and its child components.

```
METADATA|GoalPlanSet|GoalPlanSetExternalId
DELETE|GoalPlanSet|GPS_006_01
```

Related Topics

- [Goal Plan Sets: Explained](#)

Loading Goal Plan Sets: Examples

This topic provides examples showing how to load goal plan sets using HCM Data Loader.

Creating a Goal Plan Set with Goal Plans

This example GoalPlanSet.dat file creates a goal plan set and includes two goal plans.

```
METADATA|GoalPlanSet|GoalPlanSetExternalId|GoalPlanSetName|Description|GoalPlanSetActiveFlag|StartDate|
EndDate|ReqSubmittedByPersonNumber|RequestSubmissionDate|ReviewPeriodName
MERGE|GoalPlanSet|GPS_002_01|GPS_002_01|GPS_002_01|A|2015/01/01|2015/12/31|8153756|2015/09/28|Default Review
Period - Vision Corporation Enterprise
COMMENT Assign two goal plans without worker or eligibility profile assignment to the goal plan set
METADATA|GoalPlanSetPlan|GoalPlanSetExternalId|GoalPlanExternalId|Weighting
MERGE|GoalPlanSetPlan|GPS_002_01|GP_001_02|60
MERGE|GoalPlanSetPlan|GPS_002_01|GP_001_03|40
```

In this example, the goal plan set has no mass request component. Therefore, the mass request record for the goal plan set is created automatically using the **ReqSubmittedByPersonNumber** and **RequestSubmissionDate** attribute values.

Creating a Goal Plan Set with a Mass Request Component

This example GoalPlanSet.dat file creates a goal plan set with a mass request component. You don't provide the **ReqSubmittedByPersonNumber** and **RequestSubmissionDate** attributes for the goal plan set component in this case.

```
METADATA|GoalPlanSet|GoalPlanSetExternalId|GoalPlanSetName|Description|GoalPlanSetActiveFlag|StartDate|
EndDate|ReviewPeriodName
MERGE|GoalPlanSet|GPS_001_02|GPS_001_02|GPS_001_02|A|2016/01/01|2016/12/31|Default Review Period - Vision
Corporation Enterprise
METADATA|MassRequest|GoalPlanSetExternalId|ReqSubmittedByPersonNumber
MERGE|MassRequest|GPS_001_02|8153756
```

Creating a Goal Plan Set with Mass Request and Mass Request Hierarchy Components

This example GoalPlanSet.dat file creates a goal plan set with mass request and mass request hierarchy components.

```
METADATA|GoalPlanSet|GoalPlanSetExternalId|GoalPlanSetName|Description|GoalPlanSetActiveFlag|StartDate|
EndDate|ReviewPeriodName
MERGE|GoalPlanSet|GPS_001_04|GPS_001_04|GPS_001_04|A|2016/01/01|2016/12/31|Default Review Period - Vision
Corporation Enterprise
METADATA|MassRequest|GoalPlanSetExternalId|ReqSubmittedByPersonNumber
MERGE|MassRequest|GPS_001_04|8153756
COMMENT Assigning goal plan set GPS_001_04 to manager and reports
METADATA|MassRequestHierarchy|GoalPlanSetExternalId|ManagerAssignmentNumber|CascadingLevel
MERGE|MassRequestHierarchy|GPS_001_04|EEEE8153786|999999
```

Loading Talent Pools: Examples

A talent pool is a group of workers. You use talent pools to manage the development of the pool members for specific purposes. For example, you can track potential candidates for a succession plan in a talent pool or create a talent review meeting based on a talent pool. This topic provides examples showing how to create and manage talent pools using HCM Data Loader.

Creating a Talent Pool

A talent pool can reference any combination of job, job family, model profile, position, department, and grade. These objects, which can define the purpose of the talent pool, have no effect on its processing. However, they must exist in the target environment before you load a talent pool that references them. This example TalentPool.dat file creates a talent pool for a job. Source keys are used to identify both the talent pool and the associated job.

```
METADATA|TalentPool|SourceSystemOwner|SourceSystemId|PoolName|PoolTypeCode|Description|
OwnerPersonId(SourceSystemId)|Status|JobId(SourceSystemId)
MERGE|TalentPool|VISION|TP_T1|Job Chief Exec|TALENT|Talent pool description|1008|A|93079
```

Always set the **PoolTypeCode** attribute to **TALENT**. You must include the **Status** attribute. If you supply no value, then the default status is **A** (Active). Set the status to **I** (Inactive) to inactivate the talent pool.

The Talent Pool component defines the talent pool and the initial pool owner. To identify additional pool owners, you use the Talent Pool Owner component. Additional pool owners are optional, but multiple owners are recommended. Talent pool owners can see talent pools in the application. Use the Talent Pool Member component to define the pool members. This example TalentPool.dat file creates a talent pool with pool members. Source keys are used to identify the talent pool and reference foreign objects.

```
METADATA|TalentPool|SourceSystemOwner|SourceSystemId|PoolName|PoolTypeCode|Description|
OwnerPersonId(SourceSystemId)|Status
MERGE|TalentPool|VISION|TP_T2|Member Talent Pool|TALENT|Talent pool description|1011|A
METADATA|PoolMember|SourceSystemOwner|SourceSystemId|PoolId(SourceSystemId)|MemberId(SourceSystemId)
MERGE|PoolMember|VISION|TP_PM_T2_1012|TP_T2|1012
MERGE|PoolMember|VISION|TP_PM_T2_1013|TP_T2|1013
```

Translating a Talent Pool Name

This example TalentPoolTranslation.dat file translates the name of an existing talent pool. It identifies the existing record by its source key.

```
METADATA|TalentPoolTranslation|SourceSystemOwner|SourceSystemId|PoolName|Language
MERGE|TalentPoolTranslation|VISION|TP_T1|Spanish Translated Pool Name|ES
MERGE|TalentPoolTranslation|VISION|TP_T1|French Translated Pool Name|FR
```

This example TalentPoolTranslation.dat file translates the name of an existing talent pool. It identifies the existing record by its user key.

```
METADATA|TalentPoolTranslation|BasePoolName|PoolStatus|PoolName|Language
MERGE|TalentPoolTranslation|Job Chief Exec|A|Spanish Translated Pool Name|ES
MERGE|TalentPoolTranslation|Job Chief Exec|A|French Translated Pool Name|FR
```

Deleting Talent Pools

You can't delete a talent pool using HCM Data Loader. However, you can:

- Make the talent pool inactive.
- Delete talent pool owners, provided that at least one pool owner remains.
- Delete Talent Pool Member components.

This example TalentPool.dat file deletes an existing Talent Pool Owner component. It identifies the pool owner record by its source key.

```
METADATA|PoolOwner|SourceSystemOwner|SourceSystemId
DELETE|PoolOwner|VISION|TP_PO_T2_3423|TP_PM_T2_1013
```

Related Topics

- [Talent Pools: Explained](#)

15 Importing and Loading Data Using HCM Data Loader

Importing and Loading Data: Points to Consider

HCM Data Loader imports data initially to stage tables, where some validation occurs. HCM Data Loader then loads valid logical objects from the stage tables to the application tables. You can perform the import and load steps either separately or as one step. This topic summarizes the processing that occurs in each step. It also describes how you specify import and load options.

Importing Records to the Stage Tables

In the import stage, HCM Data Loader:

- Extracts the business-object data from your .zip file on the Oracle WebCenter Content server and decrypts it, if the file is encrypted.
- Validates the business-object file name and METADATA definitions and raises any errors.
- Imports records from the data files into the HCM Data Loader stage tables.
- Validates the data against attribute data types and raises any errors.
- Groups the data by local key values to form logical records of related date-effective file lines. For example, all date-effective file lines supplied for a job object are grouped into a logical occurrence of a job, such as the Accountant job.
- Forms logical objects by resolving references to parent components. For example, the logical record for a valid grade is associated with its parent job object.

Loading Objects to the Application Tables

HCM Data Loader doesn't load valid logical objects directly into the application tables. Instead, it passes valid object data to business-object-specific services. For example, to load worker objects, HCM Data Loader passes them to the worker service. These services perform the validation that's specific to the business object, raise any errors, and load valid objects to the application tables.


Setting the File Action Parameter

When you import a file on the Import and Load data page, you can set the **File Action** parameter on the Schedule Request page. The value that you specify here overrides the enterprise default setting. This table describes how the **File Action** parameter controls import and load.

File Action Value	Description
Import and load	HCM Data Loader imports the contents of the .zip file to the stage tables and calls object services to load valid logical objects to the application tables.

File Action Value	Description
Import only	HCM Data Loader imports the contents of the .zip file to the stage tables but doesn't pass valid logical objects to the relevant object service. Instead, you perform that step separately on the Import and Load data page by selecting the Load action for each business object.

You may want to select **Import only** when you start to load data to enable you to fix import errors before attempting to load objects. Once your data loading is routine, the **Import and load** option is more appropriate.

 **Tip:** You can validate your data file before you load it. Use the Data File Validator tool for HCM Data Loader. For more information, see Data File Validator Tool for HCM Data Loader (2022617.1) on My Oracle Support at <https://support.oracle.com>.

Automating Data Loading

When you automate data loading using the HCM Data Loader web service, you can set **File Action** to **Import only** or **Import and load**. If you select **Import only**, then you load the imported objects manually on the Import and Load Data page.

Reviewing Object History

For some objects, such as worker assignment, you can review the object history to see who last updated the object. When the object was last updated using HCM Data Loader, the **Last Updated By** value is FUSION_APPS_HCM_ESS_LOADER_APPID. You don't see the user name of the user who uploaded the changes.

Importing and Loading Data: Procedure

The file of data that you're preparing to import and load using HCM Data Loader may exist locally on your desktop. Alternatively, you may already have loaded it to the Oracle WebCenter Content server. This topic provides steps for importing and loading your data interactively on the Import and Load Data page. It also describes how to load data to the application tables if you imported it separately to the stage tables.

Importing and Loading a Local File

Follow these steps:

1. Select **Navigator - Data Exchange** to open the Data Exchange work area.
2. On the Tasks panel tab, select **Import and Load Data** to open the Import and Load Data page.
3. Select **Import File - Import Local File**.
4. In the **Select File** dialog box, click **Browse** to search for and select your file. Alternatively, drag the file from your local file browser to the **Browse** button.
5. Click **Submit** to upload your file to the Oracle WebCenter Content server. The file is loaded automatically to the HCM Data Loader import account hcm/dataloader/import and allocated a unique content ID.
6. On the Schedule Request page, review the parameter values and update them as required. Most of these values are set for the enterprise on the Configure HCM Data Loader page and can be left unchanged. The parameters are:

Parameter	Description
File Name	Name of the file on the WebCenter Content server.
Content ID	WebCenter Content ID for the file on the WebCenter Content server.
File Action	<p>The Import only option imports the business object data into the stage tables but doesn't load it to the application tables. You can start the load process manually from the user interface.</p> <p>The Import and load option automatically loads to the application tables any business object that loads successfully to the stage tables.</p>
File Encryption	Encryption type for the file.
Delete Source File	Indicates whether the source file is to be purged from the WebCenter Content server after processing.
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.
Maximum Percentage of Load Errors	Percentage of business-object instances in error that can occur for a business object before the load process stops.
Maximum Concurrent Threads for Load	The maximum number of concurrent process threads to use for loading your data set.
Load Group Size	The number of objects to process at a time on each thread. The record counts are updated only when processing completes for each group.



Note: This parameter appears on the Schedule Request page only if the **Enable Load Group Size Calculation** configuration parameter is set to **No**. Otherwise, the load-group size is calculated automatically.

7. Click **Submit**. A confirmation message showing the process ID appears.
8. Note the process ID and click **OK** to close the **Confirmation** dialog box.

Importing a File from the Oracle WebCenter Content Server

You can load a .zip file to the Oracle WebCenter Content Server if you prefer. Select **Navigator - Tools - File Import and Export** and upload the file to the hcm/dataloader/import account. To import and load a file that's already on the WebCenter Content server, follow these steps:

1. On the Tasks panel tab in the Data Exchange work area, select **Import and Load Data** to open the Import and Load Data page.
2. On the Import and Load Data page, click **Import File**.

In the **WebCenter Content Files** dialog box, any files in the hcm/dataloader/import account that haven't been processed are listed. You can filter the list to reduce it. For example, you can enter the file creation date.


3. Select your file in the **WebCenter Content Files** dialog box and click **Submit**.

4. On the Schedule Request page, review the parameters, edit them as necessary, and click **Submit**.

Starting or Resubmitting a Load

You can start the load process directly for a business object. You may want to do this because you submitted your data set in **Import only** mode. Alternatively, you may want to resubmit a failed load process. Follow these steps:

1. In the Business Objects table on the Import and Load Data page, select the business object and click **Load**.

 **Note:** The **Load** button is enabled only for business objects with a **Loaded** status of **Ready**, **Error**, or **Stopped**.

2. On the Schedule Request page, review the parameter values, edit them if necessary, and click **Submit**.

Import and Load Data Search Results

This topic describes the values in the search results on the Import and Load Data page. When you open the page, it contains details of any import and load processes that you ran during the last 7 days. You can enter search values to update the search results. Use the information on this page to monitor the progress of your data loads.

To open the page, select **Navigator - My Workforce - Data Exchange**. In the Tasks panel tab of the Data Exchange work area, select **Import and Load Data**. Not all values described in this topic appear by default. Select **View - Columns** to manage column display.

Values in the Import and Load Data Search Results

Data Set

Name of the compressed data file or spreadsheet containing business objects to be imported and loaded.

File Type

File type of the data set. Is either **Compressed DAT File** or **Spreadsheet**.

Integration Type

Identifier of the associated integration type when the file is loaded from another application. Otherwise, this value is blank.

Content ID

Oracle WebCenter Content identifier for the data set.

Progress

Summary of the import and load stages for logical objects.

Value	Description
Imported	Status of logical object import.

Value	Description
Loaded	Status of logical object load.
Messages	Errors and warnings that occurred during either import or load.

Imported Physical Rows

Records imported to the HCM Data Loader stage tables.

Value	Description
Imported (%)	Percentage of records in the data set that have been imported.
Total	Number of records in the data set.
Complete	Number of records imported either successfully or with warnings.
Failed	Number of records that failed to import. Positive numbers in this column link to more information about the failures.

Objects

Objects imported to the application tables.

Value	Description
Loaded (%)	Percentage of objects in the data set that loaded either successfully or with warnings.
Total	Number of logical objects in the data set.
Pending	Number of logical objects imported to the stage tables but not yet loaded.
Complete	Number of logical objects loaded either successfully or with warnings.
Failed	Number of logical objects that failed during either import or load. As objects are constructed during the import phase, you may see failed objects before the load starts.
Corrected	Number of logical objects corrected manually on the Error Management page.

Created By

User name of the worker who imported and loaded the data set.

Process ID

Process ID of the import and load process. You can see process details on the Processes tab in the Details section of the page for a selected data set.

Creation Date

Date when the data set was created.

Last Updated Date

Date when the data set was last updated.

Uploading Data Using HCM Data Loader: Worked Example

This example shows how to import and load objects manually to Oracle HCM Cloud. Import and load are performed separately.

The following table summarizes key decisions for this scenario.

Decisions to Consider	In This Example
Which business object am I loading?	Job. The child components of the job object aren't being loaded. These components are: <ul style="list-style-type: none"> • Job evaluation • Job extra information • Job legislative extra information • Job valid grade
What's the template name?	Job.dat
What's the name of the file that I am uploading?	Sales_Jobs.zip
What's the value of the File Action parameter?	Import only. You have decided to manage the load step separately.

Summary of Tasks

Upload the jobs data by:


1. Generating the jobs template
2. Entering data in the jobs template
3. Importing jobs to the stage tables
4. Loading jobs to the application tables

Generating the Job Template

1. Select **Navigator - My Workforce - Data Exchange** to open the Data Exchange work area.
2. On the Tasks panel tab, click **Initiate Data Load**.
3. On the Initiate Data Load page, select the row for job in the Search Results section.

 **Tip:** To locate the row, enter **Job** in the **Business Object** field of the Search section and click **Search**.

4. Click **Generate Template** on the table toolbar.
5. On the Schedule page, click **Submit**.
6. Click **OK** to close the confirmation message.
7. Click **Refresh** on the table toolbar to view the **Download File** icon in the **File** column for the job row.
You may have to refresh more than once before the icon appears.

 **Tip:** If the icon already exists, then you can download the existing template. The **Last Generated** column shows when the template was generated.

8. Click the **Download File** icon.
9. Save the Job.dat template locally.

Entering Data in the Job Template

1. Open the Job.dat template.
2. Edit the template file as follows:
 - o Remove METADATA lines for these unwanted child components: job evaluation, job extra information, job legislative extra information, and job valid grade.
 - o Edit the job METADATA line to remove unwanted attributes.
 - o Make any required changes to the attribute order.

3. Enter jobs data in the configured template. For example:

```
METADATA|Job|SourceSystemOwner|SourceSystemId|EffectiveStartDate|SetCode|JobCode|Name|ActiveStatus|
FullPartTime|RegularTemporary|JobFamilyId(SourceSystemId)|JobFunctionCode|ManagerLevel
MERGE|Job|VISION|SALES_CONS|2000/01/01|COMMON|SALES_CONS|Sales Consultant|A|FULL_TIME|R|CONSULT||9
MERGE|Job|VISION|SALES_MGR|2000/01/01|COMMON|SALES_MGR|Sales Manager|A|FULL_TIME|R|MANAGEMENT|MANAG|6
MERGE|Job|VISION|SALES_DIR|2000/01/01|COMMON|SALES_DIR|Sales Director|A|FULL_TIME|R|MANAGEMENT|MANAG|1
MERGE|Job|VISION|MRKT_CONS|2000/01/01|COMMON|MRKT_CONS|Marketing Consultant|A|FULL_TIME|R|CONSULT||9
MERGE|Job|VISION|MRKT_MGR|2000/01/01|COMMON|MRKT_MGR|Marketing Manager|A|FULL_TIME|R|MANAGEMENT|MANAG|6
MERGE|Job|VISION|MRKT_DIR|2000/01/01|COMMON|MRKT_DIR|Marketing Director|A|FULL_TIME|R|MANAGEMENT|MANAG|1
```

4. Save and close the template file.
5. Compress the template file. Name the compressed file Sales_Jobs.zip and save it locally.

Importing Data to the Stage Tables

1. On the Tasks panel tab in the Data Exchange work area, click **Import and Load Data**.
2. On the Import and Load Data page, select **Import File - Import Local File**.
3. In the **Select File** dialog box, click **Browse**.

4. Select the Sales_Jobs.zip file.
5. Click **Submit**.
6. On the Schedule Request page, select **Import only** in the **File Action** field.
7. Click **Submit**.
8. Click **OK** to close the confirmation message.
9. On the Import and Load Data page, click **Refresh** in the **Search Results** table.
10. Review the status of the Sales_Jobs.zip data set that you uploaded.

When the **Imported** column shows **Success** and the **Imported(%)** column displays **100**, you're ready to load the data to the application tables.

Loading Data to the Application Tables

1. In the Search Results table on the Import and Load page, select the Sales_Jobs.zip data set.
 2. In the Details table on the Import and Load Data page, select the job row and click **Load**.
 3. In the **Schedule Request** dialog box, click **Submit**.
 4. Click **OK** to close the confirmation message.
 5. Click **Refresh** in the Details table toolbar to check the status of the load.
- When the **Loaded** column shows **Success** and the **Loaded (%)** column displays 100, the job data has successfully loaded to the application tables.


Reviewing and Correcting Errors: Procedure

Errors that occur during the import and load stages of data loading are reported automatically on the Import and Load Data page. This topic describes how to review and correct reported errors.

Reviewing Physical Row Errors

The **Failed** column for **Imported Physical Rows** shows the number of records that failed to import. Follow these steps to review the errors when the **Failed** column contains a positive number:

1. Click the number in the **Failed** column to open the File Line Errors page.

 **Tip:** You can also open this page from the **Number of Messages** column on the Messages page. You open the Messages page when you select a message in the **Messages** column on the Import and Load Data page.

2. In the File Structure section of the File Line Errors page, expand any folders to see file lines with import errors.
3. When you select a line with errors in the File Structure section of the page:
 - The Messages section of the page shows the associated error message.
 - The File Line tab in the Details section of the page shows the contents of the physical row.Using this information, you can determine how to correct the import errors.

Examples of physical row errors are:

- The instruction or discriminator isn't recognized.


- The number of values in a data line doesn't match those defined in the METADATA line.
- A unique reference for the component wasn't supplied.
- The parent of a child component wasn't found, either in the file or in Oracle HCM Cloud.
- A required attribute is missing.

You must import and load the corrected data again, and you must add the corrected data file to a new compressed data file. You can't process the existing file again.

Reviewing Object Errors

The **Failed** column for **Objects** shows the number of objects that failed to load. Follow these steps to review the errors when the **Failed** column contains a positive number.

1. Click the number in the **Failed** column to open the Object Errors page.

 **Tip:** You can also open this page from the **Number of Messages** column on the Messages page. You open the Messages page when you select a message in the **Messages** column on the Import and Load Data page.

2. In the Data Structure section of the Object Errors page, expand any folders to see objects with errors. When you select an object with errors, the Details section of the Object Errors page shows:
 - On the Attributes tab, the object attributes and the values that you supplied for those attributes.
 - On the Source System Reference tab, any source-system references supplied in the METADATA and data lines in your file. Source system references enable you easily to identify the data source.
 - On the File Line tab, the contents of the physical row.

The Messages section of the Object Errors page displays error messages for the selected object. Using this information, you can determine how to correct the errors.

Correcting Objects in Error

If you prefer to correct object errors in the source data, then you must import and load the corrected data again. You must add the corrected data file to a new compressed data file. You can't process the existing file again. Alternatively, you can correct individual attribute values for objects that are in error on the Object Errors page. Follow these steps:

1. In the Data Structure section of the Object Errors page, select the record that you want to correct.
2. On the Attributes tab in the Details section, click **Edit**.
3. Enter a value in the **Current Value** column for each attribute that you're correcting. To reset any corrected attribute value to the value supplied in the file, select the attribute and click **Reset**.
4. Save your changes.

You can resubmit the load process for corrected objects on the Import and Load Data page. You can also see how many objects you corrected in the **Corrected** columns of the tables on the Import and Load Data page. When you click the number in the **Corrected** column, the Object Errors page opens showing relevant corrections. Corrected records and attributes are identified by the **Updated** icon, which is a blue sphere.

Stopping HCM Data Loader Processing: Procedure

You can stop the processing of either a data set or an individual business object. This topic explains how to stop processing.

Stopping the Processing of a Data Set

Follow these steps:

1. In the Search Results section of the Import and Load Data page, select the data set that you want to stop.
2. Click **Refresh** to update the status and confirm that the data set is still processing.
3. Click **Stop**.

 **Tip:** The **Stop** button is active only if either the import or the load is still in progress.

4. In the **Warning** dialog box, click **Yes** to confirm the stop request.

The status of the import or load process that was stopped is updated to **Stopped**. A large data set may take a few minutes to stop processing completely. For efficiency in import and load processing, HCM Data Loader checks at intervals for stop requests.

Stopping the Processing of a Business Object

Follow these steps:

1. In the Details section of the Import and Load Data page, select the business object that you want to stop processing.
2. Click **Refresh** to update the status and confirm that the business object is still processing.
3. Click **Stop**.
4. In the **Warning** dialog box, click **Yes** to confirm the stop request.

The status of the import or load process that was stopped is updated to **Stopped**.

Restarting Stopped Processes

You can't restart a stopped import process for a business object. However, you can restart a stopped load process for a business object. Select the stopped business object in the Details section of the Import and Load Data page and click **Load**. The status of both the business object and the associated data-set load processes changes to **In Progress**.

Automating Data Loading: Overview

You can automate HCM Data Loader processes to load data to Oracle HCM Cloud and extract both status and error information. This topic provides an introduction to the automation processes.

The processes that you can automate are:

- Delivery of .zip files to the Oracle WebCenter Content server

- Initiation of HCM Data Loader to process files on the WebCenter Content server
- Initiation of the HCM Data Loader Data Set Summary extract and retrieval of its output

Complete instructions are provided in Automating HCM Data Loader (1664133.1) on My Oracle Support at <https://support.oracle.com>.

Delivering Files to Oracle WebCenter Content Automatically

To automate the delivery of .zip files to the WebCenter Content server, you can use:

- The WebCenter Content Document Transfer Utility, a feature-rich Java library that provides content import and export capabilities
- WebCenter Content Generic SOAP Web Service, which exposes a SOAP web service interface (GenericSoapPort) that supports various operations, including file upload

Initiating HCM Data Loader Automatically

You can initiate HCM Data Loader automatically to process multiple files on the Oracle WebCenter Content server using the HCM Data Loader Integration Service. This web service provides two methods:

Method	Description
importAndLoadData	Initiates the processing of a data file
getDataSetStatus	Returns the current status of a data file

Several ways exist of running the web service. For example, you can use generated proxy classes.

Initiating the HCM Data Loader Data Set Summary Extract Automatically

HCM Extracts is a tool for reporting and outbound integrations. When you use HCM Extracts as part of an integrated process, you can initiate it automatically from an automated flow. The Payroll Flow Engine is the generic processing engine that you use for this purpose.

When you save a copy of the predefined HCM Data Loader Data Set Summary extract, you also create a payroll flow of type Extract. You can initiate this payroll flow using the FlowActionsService, which you use to automate and monitor the extract. You use the payrollProcessingActionService.fetchExtractOutput method to download the extract output. When the extract is associated with a BI template, the payroll flow ensures that the BI report is generated and written to the BI server.

Related Topics

- [Automating HCM Data Loader \(1664133.1\)](#)
- [HCM Data Loader Extracts: Explained](#)

FAQs for Importing and Loading Data Using HCM Data Loader

What's an HCM Data Loader data set?

A set of data supplied together to be loaded into Oracle HCM Cloud. The source of a data set is either a compressed data file containing delimited business-object-specific data files or an HCM Spreadsheet Data Loader spreadsheet.

What happens in the import stage?

The application imports individual file lines from your compressed data file to the HCM Data Loader stage tables. Related file lines are grouped to form logical business objects, and some preliminary validation occurs. Errors reported at this stage are specific to the object itself. The object's dependencies on other objects aren't assessed.

What happens in the load stage?

The application validates foreign-object references and applies business-object-specific validation rules. If the object passes all validation, then it's loaded to the Oracle HCM Cloud application tables. If any component of the object fails validation, then the whole logical object fails to load. You can monitor object status on the Import and Load Data page. Errors reported at this stage may relate to either the object itself or its dependencies on other objects.

How can I view the data in error for a data set or business object?

Click the number in the relevant **Failed** column on the Import and Load Data page. Alternatively, click the number in the **Number of Messages** column on the Messages page. Numbers greater than zero in these columns become active links. The Error Management page that opens has different views based on how you navigate to it. If you navigate from the **Failed** column for imported physical rows, then you can review the file lines in error. If you navigate from the **Failed** column for objects, then you can review the objects in error.

How do I fix errors?

You must fix errors identified during import to the stage tables in the source data. Upload the corrected data again in a new compressed data file. You can't process the existing file again.

You can also fix errors identified during load to the application tables in the source data and upload the corrected data again. Alternatively, you can correct individual errors on the Object Errors page. In this case, you resubmit the load process for corrected objects on the Import and Load Data page.

16 Data Loading Using Spreadsheets

Spreadsheet Data Loaders: Explained

This topic introduces the various spreadsheet data loaders provided with Oracle Fusion HCM. To use any of the spreadsheet data loaders, you must install the Oracle ADF Desktop Integration Add-in for Excel.

HCM Spreadsheet Data Loader

HCM Spreadsheet Data Loader is a generic data loader that you can use to load most objects that HCM Data Loader supports. The exceptions are objects, such as attachments and person images, that don't provide user keys. Objects are loaded from the spreadsheet to the HCM Data Loader stage tables. HCM Spreadsheet Data Loader is available in the Data Exchange work area.

Payroll Batch Loader

Payroll Batch Loader is a generic data loader that supports creation of several payroll objects, including elements, element entries, external bank accounts, and payroll definitions. Using Payroll Batch Loader, you can also update or delete several of the supported objects. Payroll Batch Loader is available in the Data Exchange, Payroll Administration, and Checklist work areas.

Single-Object Spreadsheet Loaders

Oracle Fusion HCM provides many spreadsheet loaders for specific object types.

In most cases, you:

- Create the spreadsheet on the page where you manage the object.
- Load the objects directly to the application tables.
- Correct any errors in the spreadsheet.

In many cases, you can both create and update the object in a spreadsheet.

Single-object spreadsheets are most useful for users such as HR specialists and benefits specialists, who are creating objects of a specific type in bulk. You're less likely to use these types of spreadsheet loaders for integrations.

This table shows some key objects that you can create in bulk using this type of spreadsheet. It also identifies the page or tab where you generate the spreadsheet.

Business Object	Page or Tab
Benefit enrollments	Evaluation and Reporting work area, Enrollment Uploads tab
Benefit plans	Manage Benefit Plans page
Benefit programs	Manage Benefit Programs page

Business Object	Page or Tab
Competencies	Manage Content Items page
Departments	Manage Departments page
Jobs	Manage Jobs page
Locations	Manage Locations page
Performance ratings	Manage Content Items page
Postal code ranges and service areas	Manage Postal Code Ranges and Service Areas page
Regulations	Manage Regulations page
Reporting groups	Manage Reporting Groups page
Stock grants	Manage Stock Grants page
Variable rate profiles	Manage Benefit Rates page
Work patterns	Manage Work Schedules page
Work schedule assignments	Manage Work Schedule Assignment Administration page
Work schedules	Manage Work Schedules page
Worker goals	Administer Goals page

For some objects, such as benefit rates, person habits, and assignments, you can perform a mass update in a spreadsheet. To update these objects, you download existing objects to a generated spreadsheet, edit the objects, and upload the changes.


Setting Up the Desktop Integration for Excel: Procedure

Desktop integrated Excel workbooks let you create or edit records that you can upload to the application. To use these workbooks, you must install a desktop client and set up Microsoft Excel.

Prerequisites

Perform these prerequisites before you install the client.

- Make sure that you have:
 - Microsoft Excel 2007, 2010, 2013, or 2016 (32-bit recommended)
 - Microsoft Windows 7, 8.1, or 10
- If you're reinstalling the client and currently have a version older than 11.1.1.7.3 (4.0.0), then uninstall that Oracle ADF Desktop Integration Add-In for Excel client the same way you uninstall any program on your computer.

 **Tip:** You can find the version in the control panel where you uninstall programs, or in the About dialog box in Excel.

- Optionally install the following from the Microsoft web site.
 - Microsoft .NET Framework 4.5.2
 - Microsoft Visual Studio 2010 Tools for Office Runtime (VSTO Runtime)

The desktop client installer does check if you have these already, and would download and install them if needed. But, you can manually install them first, especially if you run into issues installing them as part of installing the desktop client.

Installing the Desktop Client

Install the Oracle ADF 11g Desktop Integration Add-In for Excel, which is a desktop client that lets you use the integrated workbooks that you download from the application.

1. Make sure you are signed in to your computer with your account. For example, you can't have someone else sign in as an administrator and make the installation available for everyone using your computer.
2. In the application, look for the client installer under **Navigator - Tools**. If it's not there, then ask your help desk where you can find the installer.
3. Run the installer (**adfdi-excel-addin-installer.exe**) as you would any program that you install on your computer.

Setting Up Microsoft Excel

Integrated workbooks can have buttons within the worksheet, which you click to perform an action, for example to upload the worksheet. If you use any workbook with such buttons, then perform the following steps in Microsoft Excel only once, even if you reinstall the desktop client.

1. Click the **Microsoft Office** button, and click the **Excel Options** button.
2. In the Excel Options dialog box, select the Trust Center tab, and click **Trust Center Settings**.
3. In the Trust Center dialog box, select the Macro Settings tab, and select the **Trust access to the VBA project object model** check box.

 **Note:** The exact steps can vary depending on your version of Microsoft Excel.

Related Topics

- [Using Desktop Integrated Excel Workbooks: Points to Consider](#)

Troubleshooting the Desktop Integration for Excel: Procedure

The application is integrated with Microsoft Excel so that, where available, you can work with records in a desktop integrated workbook. You might run into issues with the integration, for example, if you can't open the workbook that you downloaded or the workbook doesn't look right.

Verifying the Version of Your Desktop Client

First, check the version of your Oracle ADF 11g Desktop Integration Add-In for Excel client.

1. Find Oracle ADF 11g Desktop Integration Add-In for Excel in the control panel where you uninstall programs on your computer, and see the **Version** column. Or, open the About dialog box in Excel, for example by doing one of the following:
 - Without opening a desktop integrated workbook, click the Microsoft Office button in Excel, select **ADF Desktop Integration**, and then select **About Oracle ADF 11g Desktop Integration**.
 - With a desktop integrated workbook open, click the Ribbon tab named after the task you're doing in the workbook, if available. Click **About**, and open the Versions tab, and see the version for the ADF Desktop Integration component.
2. Ask your help desk if you need to reinstall the client, due to the version you're on.

Using the Client Health Check Tool

Use the health check tool to find out what integration issues you might have and how to resolve them. Ask your help desk if you are unable to find or use the tool.

1. Open a desktop-integrated workbook, and note down the URL in the Connect dialog box.
2. In a web browser, open the URL and click the **Run client health check tool** link to download the tool.
3. Run **ClientHealthCheck.exe** as you would other programs on your computer, and review the result for each checked item.
4. Select any item that has a problem, and read the help text.
5. Fix some of the problems by clicking the **Fix Problems** button. Otherwise, follow the instructions in the help text.
6. If you need more assistance, click the **Save Report As** button to prepare information for your help desk.
7. Review the report and remove any sensitive information.
8. Contact your help desk and provide your report.

Reenabling the Add-In

In some cases, the desktop integration add-in for Excel can become disabled. If you have the right version installed, but the add-in isn't working and you're not using the health check tool, then try the following:

1. Click the **Microsoft Office** button, and click the **Excel Options** button.
2. Select the Add-Ins tab, and check which list the **Oracle ADF 11g Desktop Integration Add-in for Excel** is in.
3. If **Oracle ADF 11g Desktop Integration Add-in for Excel** is in the **Disabled Application Add-ins** list:
 - a. Select **Disabled Items** in the **Manage** field and click **Go**.

- b.** In the Disabled Items dialog box, select **Oracle ADF Desktop Integration Add-In for Excel** and click **Enable**.
 - c.** Back in the Excel Options dialog box, select **COM Add-ins** in the **Manage** field and click **Go**.
 - d.** In the COM Add-Ins dialog box, select **Oracle ADF 11g Desktop Integration Add-in for Excel**.
- 4.** If **Oracle ADF 11g Desktop Integration Add-in for Excel** is in the **Inactive Application Add-ins** list, then perform only steps 3.3 and 3.4.
- 5.** If **Oracle ADF 11g Desktop Integration Add-in for Excel** is in the **Active Application Add-ins** list, then contact your help desk.

 **Note:** The exact steps can vary depending on your version of Microsoft Excel.

17 HCM Spreadsheet Data Loader

Using HCM Spreadsheet Data Loader: Points to Consider

You can load data to the HCM Data Loader stage tables from either a data (.dat) file or a spreadsheet. Your approach depends largely on how often you load data and the complexity of the data that you're loading. This topic describes some key features of data loading from spreadsheets. This information helps you to identify the best approach for your business case.

Spreadsheet Templates

You must generate the spreadsheet from a spreadsheet template. Some spreadsheet templates are predefined. To create custom templates, you can either copy a predefined template and edit it or create a template from scratch.

When to Use Spreadsheets

Spreadsheets offer many advantages. For example:


- Spreadsheets are easy to use.
- You can create custom spreadsheet templates and generate spreadsheets tailored to your business case. For example, you can omit attributes, change attribute order, change attribute labels, and add help text.
- You can save data periodically from a spreadsheet to the stage tables. Therefore, you can load large volumes of records without being prevented by limits on spreadsheet size.
- Errors are reported in the spreadsheet, and you can correct them there.
- You can mix your use of spreadsheets with use of .dat files. For example, you can load data from a spreadsheet but maintain it using .dat files.
- In specific cases, spreadsheets are the better option. For example, to load an element entry for an element with 5 lines from a .dat file, you supply:
 - 1 record for the element entry
 - 1 record for each name and value

In a spreadsheet, you supply just 1 spreadsheet line where the values are individual columns on the same line.

Spreadsheet Restrictions

When loading data from a spreadsheet:

- You can create or update business object components but you can't delete them.
- You can't load attachments or images.
- HCM Spreadsheet Data Loader doesn't recognize source keys. Therefore, objects that you load from a spreadsheet have default source keys. To maintain such objects using .dat files, you supply user keys.

 **Note:** Whether you load data from a spreadsheet or a .dat file, no approval processes exist for loaded objects.

HCM Data Loader Scope

You can use HCM Spreadsheet Data Loader to import data to the HCM Data Loader stage tables only if the **HCM Data Loader Scope** parameter is set to **Full**. This parameter has been set to **Full** by default for all new customers since Release 10 and can't be changed.

Setup Requirements

All users must set up desktop integration for Microsoft Excel to use HCM Spreadsheet Data Loader.

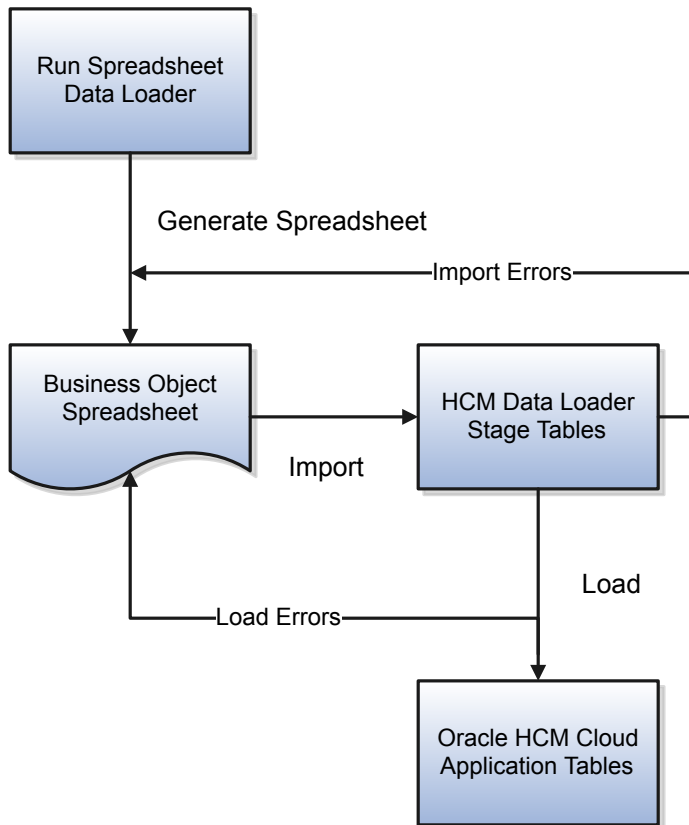
Related Topics

- [HCM Data Loader Scope: Explained](#)
- [Setting Up the Desktop Integration for Excel: Procedure](#)

Uploading Data Using HCM Spreadsheet Data Loader: How It's Processed

When you click Upload in a spreadsheet, your data is uploaded to Oracle HCM Cloud. Behind the scenes, upload is a two-stage process. In the first stage, HCM Spreadsheet Data Loader imports the data to the HCM Data Loader stage tables. In the second stage, HCM Data Loader loads valid business objects to the application tables.

This figure summarizes the process:




How Spreadsheet Data Is Loaded

The details of this process are:

1. You perform the Run Spreadsheet Data Loader task in the Data Exchange work area to generate a spreadsheet for a business object from a spreadsheet template.
2. You click **Create Data Set** in the spreadsheet to create a data set for the data that you want to load.
3. You enter data in the spreadsheet and click **Upload**. This action imports the spreadsheet data set to the HCM Data Loader stage tables.
4. Some validation occurs in the stage tables. Any import errors are reported in the **Progress** and **Message** columns of the **Spreadsheet Line Status** section of the spreadsheet when you click **Refresh**.

Errors at this stage are rare, as the spreadsheet format enforces business-object structure and rules.

5. As each object loads successfully to the stage tables, HCM Data Loader calls the relevant object service to load the object to the application tables.
6. The object service validates the object. Any errors are reported in the **Progress** and **Message** columns of the **Spreadsheet Line Status** section of the spreadsheet when you click **Refresh**.
7. You correct any import or load errors in the spreadsheet and click **Upload** again. This action imports the corrected data to the stage tables again.

 **Tip:** You can review the status of an HCM Spreadsheet Data Loader data set on the HCM Data Loader Import and Load Data page. Search for data sets by file type to list those loaded from spreadsheets.

The Messages worksheet of the spreadsheet holds a summary of all import and load messages generated for this data set.

HCM Spreadsheet Data Loader Access Requirements

This topic identifies the function security privileges that you must have to use HCM Spreadsheet Data Loader. The predefined Human Capital Management Integration Specialist job role has all of these privileges. You can also assign them to custom roles, if appropriate.

This table describes the function security privileges that you must have to perform these tasks in the Data Exchange work area:

- Run Spreadsheet Data Loader
- Manage Spreadsheet Templates


Privilege Name and Code	Privilege Description	Run Spreadsheet Data Loader	Manage Spreadsheet Templates
Access HCM Spreadsheet Data Loader Templates (HRC_ACCESS_HSDL_TEMPLATES_PRIV)	Provides access to the Run Spreadsheet Data Loader task and page where users can review the active spreadsheets that they're authorized to access.	Yes	No
Load Data using HCM Spreadsheet Data Loader (HRC_LOAD_DATA_USING_HSDL_PRIV)	Allows users to generate a spreadsheet for data entry and submit the HCM Data Loader processes that upload the data from the spreadsheet.	Yes	Yes
Manage Data Exchange Work Area (HRC_MANAGE_DATA_EXCHANGE_PRIV)	Allows users to access the Data Exchange work area to perform the HCM Spreadsheet Data Loader tasks.	Yes	Yes
Manage HCM Spreadsheet Data Loader Templates (HRC_MANAGE_HSDL_TEMPLATES_PRIV)	Provides access to the Manage Spreadsheet Templates task and page where users can create and maintain spreadsheet templates.	No	Yes

HCM Spreadsheet Data Loader Templates: Explained

HCM Spreadsheet Data Loader loads HCM business objects from spreadsheets, which you generate from spreadsheet templates. A spreadsheet template for a business object or business object component contains selected attributes in a specified order. Every spreadsheet generated from the template has the same structure and organization as the template. This topic describes spreadsheet templates.

Predefined Spreadsheet Templates

Predefined spreadsheet templates may exist for commonly loaded objects, such as worker, job, and location. These templates contain columns for all supported components and attributes. Use the saved search to retrieve predefined spreadsheet templates on the Manage Spreadsheet Templates page.

 **Tip:** Copy the predefined templates and edit the copies to suit business requirements.

Spreadsheet Template Status

The status of a spreadsheet template determines whether you can edit it, delete it, or generate spreadsheets from it. This table summarizes the template status values.

Spreadsheet Template Status	Edit	Delete	Generate Spreadsheets
Draft	Yes	Yes	No
Active	Yes See note.	No	Yes
Inactive	Yes	No	No

 **Note:** You can disable attributes in an active spreadsheet template but you can't remove them.

To make an active spreadsheet template unavailable to users, you change its status to **Inactive**. You can't change it back to **Draft**.

Spreadsheet Template Features

As spreadsheet templates are highly customizable, you can create spreadsheet templates to suit business requirements. For example, you can:

- Select columns to include.
- Specify column order and headings.
- Enable and disable columns.
- Provide default values.

- Make optional values required.
- Supply help text to appear on hover on column headings.

Lists of values appear for relevant attributes.

Managing Spreadsheet Templates

To create a spreadsheet template, you can:

- Copy an existing spreadsheet template and edit the copy to create a custom template.
- Create a spreadsheet template from scratch.
- Import a spreadsheet template that was exported from a different environment.

Translating Spreadsheet Templates

You can provide translated versions of these text entries in a spreadsheet template:

- The template name and description
- Template instructions
- Column headings and descriptions

Set the session language to the target language before editing the spreadsheet template to enter translated text.


Defining Categories for Spreadsheet Templates: Procedure

If you define many spreadsheet templates, then you may want to group them by category to make retrieval easy. No categories are predefined. This topic describes how to define template categories.

Defining Template Categories

Follow these steps:

1. Select **Navigator - Setup and Maintenance**.
2. In the Setup and Maintenance work area, search for and select the **Manage Common Lookups** task.
3. On the Manage Common Lookups page, search for the lookup type **ORA_HRC_DL_TEMPLATE_CATEGORY**.
4. In the search results, select **ORA_HRC_DL_TEMPLATE_CATEGORY** to display its lookup codes.
5. Click the **New** icon to add a row to the lookup codes table.
6. Define the lookup codes for your categories. For example, to create a category for worker templates:
 - a. Enter **WORKER_TEMPLATES** in the **Lookup Code** field.
 - b. Enter **Worker** in the **Meaning** field. This value appears in the list of categories on the Manage Spreadsheet Templates page, for example.

 **Tip:** The **Display Sequence** value is a number that specifies the position of this category in the list of categories.

7. Click **Save and Close**.

You can now select your category on the Manage Spreadsheet Templates and Manage Template Definition pages. You can also search by category when you perform an advanced search for a spreadsheet template on the Load Spreadsheet Data page.

Creating and Editing Spreadsheet Templates: Procedure

Creating a spreadsheet template is a four-step process. You:

1. Define general information for the template.
2. Design the template's contents and layout.
3. Specify template instructions, if required.
4. Review the design and generate a test spreadsheet from the template.

This topic provides instructions for the four steps of this process and describes how to edit an existing spreadsheet template.

Defining the Template

Follow these steps:

1. Select **Navigator - My Workforce - Data Exchange** to open the Data Exchange work area.
2. On the Tasks panel tab, select **Manage Spreadsheet Templates** to open the Manage Spreadsheet Templates page.
3. Click **Create**.
4. In the **Create Template** dialog box:
 - a. Enter a name for the template. By default, generated spreadsheets have the name of the template with a date and time stamp.
 - b. The **Code** value is generated automatically based on the **Name** value. You can overwrite the code.
 - c. Select the business object to which the template relates.
 - d. Select the legislative data group, if the template is legislation-specific.
 - e. Enter a description and select a category, if required.
 - f. Click **OK**.

The Manage Template Definition page opens. Some fields contain values that you entered in the **Create Template** dialog box.

5. The **Status** is **Draft** by default. You can generate spreadsheets from **Active** templates only. Leave the template in **Draft** status until it's ready for use.
6. Click the Design tab. The Design page opens.

Designing the Template

The Available Attributes panel lists all attributes for the top-level component of the object. Use the **Available Attributes** menu to select a different component.

Follow these steps:

1. Drag and drop attributes from the Available Attributes panel to the Selected Attributes panel.
2. Reorder the attributes in the Selected Attributes panel to specify the order you require in generated spreadsheets.
3. Either edit individual attributes, as required, or manage all attributes at once by selecting **Configure Advanced Settings**.

4. When the template design is complete, click the Instructions tab to enter template instructions. Otherwise, click the Review tab.

Entering Template Instructions


On the Instructions page, you can enter instructions to appear on a separate Instructions worksheet in spreadsheets generated from the template. To enter instructions:

1. Click **Add**.
2. In the **Create Instruction** dialog box, enter a heading and the instruction text.
3. Repeat steps 1 and 2 to create additional sections of instruction text, as required. You can reorder and delete sections.
4. Click the Review tab. The Review page opens.

Reviewing the Template

On the Review page, your template design appears. To generate a sample spreadsheet:

1. Click **Preview**.

 **Tip:** You can click **Preview** on any page to generate a sample spreadsheet. However, the spreadsheet shows only the configuration that you have saved so far.

The default file name is **GenericHdlSpreadsheet.xlsx**.

2. Either save the file locally or open it. When you open it, you're prompted to sign in.
3. Review the spreadsheet.
4. Save the template.

When you're ready to make the template available, edit it and set its status to **Active**.

Editing the Template

To edit a spreadsheet template, search for it on the Manage Spreadsheet Templates page and click its name in the search results. You can edit an existing spreadsheet template in any status, but you can't change its code or type. In addition, you can't remove attributes from an active template, but you can disable them.

Designing Spreadsheet Templates: Explained

Creating a spreadsheet template is a four-step process. You:

1. Define general information for the template.
2. Design the template's contents and layout.
3. Specify instruction text, if required.
4. Review the design and generate a test spreadsheet from the template.

This topic describes the design step of this process.

Available Attributes Panel

The Available Attributes panel lists all attributes of all components of the object and any associated translation objects. You can:

- Switch to a tree view to see the component hierarchy. For each component of the object, you see separate entries in the hierarchy for attributes, flexfield attributes, and child components.
- Select a specific component. For example, for the location object, you could select **Location Other Address** to exclude other location components from the list of available attributes.
- Click an attribute name to view the attribute details.
- Move attributes from the Available Attributes panel to the Selected Attributes panel. When you move the first attribute, any required attributes for the component move automatically to the Selected Attributes panel. Attributes that are required conditionally don't move. You must add them explicitly to include them.


Selected Attributes Panel

In the Selected Attributes panel, attributes appear in the order in which you added them. You can:

- Change the attribute order. Click the **Reorder** icon to open the **Reorder Attributes** dialog box.
- Duplicate a component. This option exists for any component that can appear more than once in an object, such as Person Phone or Location Legislative Extra Information.

 **Tip:** Edit the selected attributes of the component before creating additional instances.

- Edit the details of an attribute. For example, you can:
 - Change the column heading.
 - Make an optional value required.
 - Specify a default value.

 **Note:** The default value doesn't appear in generated spreadsheets. To avoid confusion, you may want to exclude attributes with default values from generated spreadsheets.

- Specify whether the attribute appears in the generated spreadsheet.
- Disable optional attributes. These attributes appear but can't be updated.
- Enter a description to appear as help text in the generated spreadsheet.
- For appropriate attributes, select a list display type.

Advanced Settings

To manage all attributes at once, click **Configure Advanced Settings**. On the Advanced Options for Attribute Definition page, you can manage:

- Basic information, flexfield attributes, and lists of values.
- Attribute keys. When you import data to a spreadsheet from a file, you can include a header. The attribute names used in the import file can be different from those in the spreadsheet. In this case, the **Attributes Keys** fields must include the names from the import file. For example, you can import a location code to a spreadsheet from a file

where it's known as the location ID. To ensure that the data imports successfully, you must set the **Attribute Key** value to **Location ID** for the **Location Code** attribute.

Configuring Default Values for Spreadsheet Attributes: Explained

When you design a spreadsheet template for bulk data loading using HCM Spreadsheet Data Loader, you can specify default values for attributes. This topic describes the ways in which you can specify default values.

Specifying Default Values for Date Attributes

These options are available for specifying default date values:

Option	Description
Specified date	Using the date picker, select a fixed date.
Load date	Use the date when the spreadsheet row is loaded.
Existing selected attribute	Select another date attribute to supply the default value. The attribute must be in the Selected Attributes section of the Design page.
Use expression	Supply a Groovy expression to derive the default value.

For example, to set the default value of a date attribute to the load date:

1. Select the attribute in the Selected Attributes section of the Design page of Manage Spreadsheet Templates and click **Edit**.
2. In the **Edit** dialog box, set **Default Value** to **Value** and select **Load date** from the list.

Specifying Default Values for Attributes Other Than Dates

These options are available for specifying default values for attributes other than dates:

Option	Description
Constant value	Enter the default value.
Existing selected attribute	Select another attribute of the same data type to supply the default value. The attribute must be in the Selected Attributes section of the Design page.
Use expression	Supply a Groovy expression to derive the default value.

Supplying Groovy Expressions

Using Groovy expressions enables you to manipulate other attribute values to provide a default value. To specify a Groovy expression to derive a default value:

1. Select the attribute in the Selected Attributes section of the Design page and click **Edit**.
2. In the **Edit** dialog box, set **Default Value** to **Value** and select **Use expression** from the list.
3. Enter the expression in the text area and click **Validate** to ensure the expression is valid.

Configuring Attribute Validation Using Value Sets: Procedure

When designing a spreadsheet template for data loading, you can configure a list of validation values for an attribute based on a custom value set. You create the value set using the Manage Value Sets task. This topic describes how to configure value-set validation once the value set exists.


Configuring Attribute Validation in a Spreadsheet Template

The validation type of the value set to be used to validate spreadsheet attribute values can be one of the following:

Validation Type	Value Subtype
Independent	Text or Translated text
Dependent	Text or Translated text
Subset	Text or Translated text
Table	

To use a custom value set to validate an attribute in a spreadsheet:

1. On the Design page of the Manage Spreadsheet Templates tab, select the attributes to include in the spreadsheet.
2. In the Selected Attributes section, click the **Advanced Options for Attribute Definition** icon.
3. In the **Advanced Options for Attribute Definition** dialog box, set the **List Type** field to **Value Set** for the attribute.

 **Tip:** Scroll right to see all fields for lists of values.

4. In the **List View Object** field for the attribute, enter the value-set name.
5. If the validation type of the value set is **Table**, then enter these values concatenated in the **List Bind Map** field:
 - The relevant parameter from the value-set definition
 - The attribute key of the attribute whose value restricts the value set

Prefix the attribute key with a colon (:) and enclose the complete value in brackets ([]). For example:

Field	Value
List Type	Value Set
List View Object	REASON_CODES_VS
List Display Attributes	
List View Criteria	
List Bind Map	[pActionCode:Worker_ActionCode]
List return Value	
List Query Criteria	


In this example:

- The parameter from the value-set definition is **pActionCode**.
 - The attribute key of the attribute that restricts the value set is **Worker_ActionCode**. This value is the attribute key of an attribute selected in the spreadsheet template. Alternatively, you could supply a constant value or the current date (adf.currentDate). To supply multiple parameters, you must separate each parameter name and value pair with a semicolon (;). For example: **[pActionCode:Worker_ActionCode;pEnabled:Y]**
6. If the validation type of the value set is **Dependent**, then enter these values concatenated in the **List Bind Map** field:
- Bind_IndependentValue
 - The attribute key of the attribute on which the value set is dependent

Prefix the attribute key with a colon (:). For example:

Field	Value
List Type	Value Set
List View Object	REGION_OF_BIRTH_VS
List Display Attributes	
List View Criteria	
List Bind Map	Bind_IndependentValue:Worker_CountryOfBirth
List Return Value	
List Query Criteria	

In this example, the attribute key of the attribute on which the value set is dependent is **Worker_CountryOfBirth**.

 **Note:** A **List Bind Map** value isn't needed for value sets whose validation type is either **Independent** or **Subset**. Other fields for lists of values aren't needed for value-set validation.

7. Save your changes in the **Advanced Options for Attribute Definition** dialog box.

Related Topics

- [Value Sets: Explained](#)

Autogenerated Attribute Values in Spreadsheet Templates: Explained

In spreadsheet templates for the Worker object you can specify that values for some attributes, such as **Person Number** and **Assignment Number**, are generated automatically. For example, on the Design page in a Worker spreadsheet template, you edit the **Person Number** attribute in the Selected Attributes panel and select **Autogenerate**. In this case, users don't have to supply the person number value in generated spreadsheets. However, so that related records can be identified and processed together, a unique identifier must still be supplied. Therefore, **Person Number** must appear in generated spreadsheets and its value must be a unique identifier. This value isn't used as the person number, which is generated on successful creation of the worker record.

To ensure that users supply this unique identifier for a logical object, these rules apply to the template definition:

- If you select the **Autogenerate** option for an attribute, then the **Required** option is selected automatically and made read-only.
- If the **Required** option is selected and the **Default Value** field is set to **None**, then the **Visible** option is selected automatically and made read-only.
- A default column heading is defined. For example, for the **Person Number** attribute the column heading is **Person Unique Identifier**. For the **Assignment Number** attribute, the column heading is **Assignment Unique Identifier**. You can change the default column heading.
- The default description of the attribute explains that a unique identifier must be supplied. You can edit this description.

Spreadsheets generated from the template include the **Person Number** or **Assignment Number** attribute, as appropriate, with its new column heading and description. For the attribute value, users must enter a unique identifier to ensure that records are loaded successfully.

Copying Spreadsheet Templates: Procedure

Copying a spreadsheet template and editing the copy is an efficient way to create a new spreadsheet template. This topic describes how to copy spreadsheet templates. You perform this task in the Data Exchange work area.

Copying a Spreadsheet Template

Follow these steps:

1. On the Tasks panel tab of the Data Exchange work area, select **Manage Spreadsheet Templates**.
2. On the Manage Spreadsheet Templates page, search for the spreadsheet template to copy.
3. Select the spreadsheet template in the search results and select **Copy** from the template Actions menu.
4. In the Copy Template dialog box:
 - a. Enter a unique name for the spreadsheet template in the **Name** field.
 - b. A **Code** value is created automatically based on the name value. You can overwrite the code value.
 - c. Select a legislative data group, if the template is legislation-specific.
 - d. Enter a description and select a category, if required.
 - e. Click **OK**.
5. Click **OK** to close the **Confirmation** dialog box.
6. On the Manage Spreadsheet Templates page, search for the new spreadsheet template.
7. In the search results, click the name of the spreadsheet template to open it for editing.

You can copy a spreadsheet template in any status, but the status of the new spreadsheet template is always **Draft**. Change the status to **Active** when the template is ready to use. Active spreadsheet templates are available on the Load Spreadsheet Data page.

Exporting and Importing Spreadsheet Templates: Procedure

You can import custom spreadsheet templates to other environments to avoid having to recreate them. For example, you can export spreadsheet templates from a test environment and import them to a production environment. This topic describes how to export and import spreadsheet templates. You perform these tasks in the Data Exchange work area.

Exporting Spreadsheet Templates

Follow these steps in the source environment:

1. On the Tasks panel tab of the Data Exchange work area, select **Manage Spreadsheet Templates**.
2. On the Manage Spreadsheet Templates page, search for and select the spreadsheet template that you want to export.
3. From the Actions menu for the spreadsheet, select **Export**.
4. In the **Opening** dialog box, select **Save file** and click **OK**.
5. In the **Enter name of file to save to** dialog box, select a location for the file and click **Save**.

The file is exported to the specified location in XML format. Deliver the file to the target environment.

Importing Spreadsheet Templates

Follow these steps in the target environment:

1. On the Tasks panel tab of the Data Exchange work area, select **Manage Spreadsheet Templates**.
2. On the Manage Spreadsheet Templates page, click **Import**.

3. In the **Import** dialog box:
 - a. Enter a unique name for the spreadsheet template. This name applies in the target environment.
 - b. The **Code** field is completed automatically based on the name value. You can overwrite the value.
 - c. If your template is legislation-specific, then select the legislative data group.
 - d. Click **Choose File** to select the template file.
 - e. Enter a description and select a category, if required.
 - f. Click **OK** in the **Import** dialog box to import the template.

Click **OK** to close the **Confirmation** dialog box.

4. On the Manage Spreadsheet Templates page, search for the imported template.

The status of an imported spreadsheet template is always **Draft**. Change the status to **Active** when the template is ready for use.

Generating Spreadsheets: Procedure

To load data using HCM Spreadsheet Data Loader, you must first generate a spreadsheet from a spreadsheet template. This topic describes how to generate a spreadsheet and save it locally. You perform this task in the Data Exchange work area.

Generating a Spreadsheet

Follow these steps:

1. On the Tasks panel tab of the Data Exchange work area, select **Run Spreadsheet Data Loader**.
2. On the Load Spreadsheet Data page, enter search terms for the template. For example, to find a template for a specific business object:
 - a. Select the business object in the **Business Object** field.
 - b. Clear the **Created By** field to see all available templates.
 - c. Ensure that **Status** is set to **Active**.

 **Tip:** You can generate spreadsheets from active templates only.

- d. Click **Search**.

 **Tip:** In the search results, click the **Template Overview** icon for a template to see a list of the attributes that it supports before you make your selection.

3. Click the name of the template that you want to use.
4. In the **OpeningGenericHdlSpreadsheet.xlsx** dialog box, select **Save File** and click **OK**.
5. In the **Enter name of file to save to** dialog box, select a location and save the spreadsheet.

You can now enter data in the spreadsheet and upload it.

Importing Data to a Spreadsheet: Procedure


To populate a spreadsheet with data for upload, you can either enter the data manually or import it from a file. Importing data from a file automatically creates a data set in the spreadsheet. Alternatively, if you search for an existing data set first, you can import data from a file to an existing data set. This topic describes how to import data to a spreadsheet from a file.

Importing Data from a File

Follow these steps:

1. If the spreadsheet isn't already open, then open it and sign in when prompted.
2. On the Spreadsheet Loader toolbar, click **Import File**. The **Import from File** dialog box opens.
3. In the **Import from File** dialog box, select one of **File Type** values shown in this table.

File Type	Description
Comma separated values	Comma-separated (.csv) file
Pipe separated values	Delimited data (.dat) file
Other file type	Text file (.txt) with a custom separator
XML file type	XML file

 **Tip:** You can download CSV and XML file templates from a spreadsheet template. Use them to load data to a spreadsheet generated from the same template.

4. For file types other than **XML file type**, indicate whether the file contains a header.

The header provides the names of the attributes for which the file contains values. In this case, the attribute order in the file doesn't have to match the spreadsheet column order. If you deselect **Header included**, then the order of the values in the file must match the order of the columns in the spreadsheet.


In an XML file, the XML tags used for the attribute values must use the attribute keys configured in the spreadsheet template.

5. Click **Browse** to select the file to import.
6. For file types other than **XML file type**, expand the Advanced File Options section. Complete the options as shown in this table for the selected file type. You can use the default values where supplied.

File Type	File Separator	Escape Indicator	Newline Indicator
Comma separated values	N/A	1 - 10 characters	1 - 10 characters
Pipe separated values	N/A	1 - 10 characters	1 - 10 characters
Other file type	1 - 10 characters	1 - 10 characters	1 - 10 characters

The **Newline Command** field shows the newline command constructed from the supplied **Escape Indicator** and **Newline Indicator** values.

7. Click **OK** to import the data from the selected file. The data is imported and a name is assigned automatically to this new data set. You can overwrite the data set name and edit any of the imported attribute values.

 **Note:** Some validation of the imported data occurs. Any errors appear automatically in the **Progress** and **Message** columns of the **Spreadsheet Line Status** section of the spreadsheet. Correct any errors before you attempt to upload the data.

Managing Spreadsheet Data Sets: Procedure

A data set is all data loaded from an instance of the same spreadsheet and identified by the same data-set name. You can load the data from a spreadsheet all at once. Alternatively, you can add data rows to an existing spreadsheet and upload them to the existing data set. This topic describes how to create and manage spreadsheet data sets.

Creating a Data Set

Follow these steps:

1. Open the generated spreadsheet.
2. In the **Connect** dialog box, click **Yes** and sign in when prompted. After a few seconds, the spreadsheet shows all attributes from the selected spreadsheet template.
3. Click the Spreadsheet Loader tab in the spreadsheet toolbar.
4. Click **Create Data Set** and click **OK** to close the dialog box. A data set name is generated automatically, based on the template name, date, and time, but you can overwrite it. The name must be unique for the spreadsheet template.
5. Click **OK** to close the confirmation dialog box.

You can either enter data manually in the spreadsheet or import it from a file.

Updating a Data Set

To retrieve an existing data set, follow these steps:

1. Either open the original spreadsheet or generate a new spreadsheet from the same template as the original spreadsheet.
2. In the **Data Set Name** field under the Search header, select the name of the data set that you want to update.
Available data sets:
 - Were uploaded from a spreadsheet that was generated from the same spreadsheet template as the current spreadsheet.
 - Still exist in the stage tables. You can't update a data set after you have deleted it from the stage tables.
3. Select a **Search Mode** value. You can retrieve either all records in the data set or only those that failed to load.
4. In the Spreadsheet Loader toolbar, click **Search**. The specified data-set records are downloaded to the spreadsheet.

You can correct failed rows and add new rows, but you can't update any row that loaded successfully. Rows that you add belong to the data set that you're updating. When you upload the data, only new and updated rows are imported to the stage tables.

Saving a Data Set

You can import a data set periodically to the HCM Data Loader stage tables by clicking **Save** on the Spreadsheet Loader toolbar. The data isn't loaded to the application tables until you click **Upload**. The **Save** action lets you:

- Import data to the stage tables over a period of time. For example, you can enter and save payroll adjustments throughout the week but upload them only at the end of the week.
- Create large volumes of data that you save regularly to a single data set. You don't have to manage those volumes all at once in the spreadsheet.
- Allow multiple users to supply data for processing in the same data set.

Related Topics

- [Maintaining the Stage Tables: Procedure](#)

Uploading Data: Worked Example

This example shows how to upload jobs for the Sales Department using HCM Spreadsheet Data Loader.

This table summarizes key decisions for this scenario.

Decisions to Consider	In This Example
Which business objects am I loading?	Jobs
Which spreadsheet template am I using?	Sales Jobs
How do I populate the spreadsheet?	Import the file Sales_Jobs.csv a comma-separated file that contains a header and uses the \n newline command.

Summary of the Tasks

Upload the jobs by:

1. Generating the spreadsheet
2. Importing job objects to the spreadsheet
3. Uploading jobs and correcting errors

Prerequisites

Before you can complete this task:

1. You must have installed the desktop client Oracle ADF 11g Desktop Integration Add-in for Excel.
2. You must have enabled the Trust Center setting **Trust access to the VBA project object model** in Microsoft Excel.

3. The Sales Jobs spreadsheet template must exist and be active.
4. The Sales_Jobs.csv file must exist.

Generating the Spreadsheet

1. Select **Navigator - My Workforce - Data Exchange** to open the Data Exchange work area.
2. On the Tasks panel tab, select **Run Spreadsheet Data Loader**.
3. On the Load Spreadsheet Data page, enter **Sales Jobs** in the **Name** field. Clear the **Created By** field and click **Search**.
4. In the search results, click **Sales Jobs**.
5. In the **OpeningGenericHdlSpreadsheet.xlsx** dialog box, select **Save File** and click **OK**.
6. In the **Enter name of file to save to** dialog box, select a location.
7. Enter **SalesJobs.xlsx** in the **File name** field and click **Save**.

Importing Job Objects to the Spreadsheet

1. Open the **SalesJobs.xlsx** spreadsheet and sign in when prompted.
2. On the Spreadsheet Loader toolbar, click **Import File**.
3. Complete the fields of the **Import from File** dialog box as shown in this table:

Field	Value
File Type	Comma separated values
Header included	Yes
File	Sales_Jobs.csv
Escape Indicator	\
Newline Indicator	n

4. Click **OK**.
The data set name is generated automatically.
5. Correct any Import File Errors reported in the **Progress** and **Message** columns of the Spreadsheet Line Status section of the spreadsheet.

Uploading Jobs and Correcting Errors

1. In the Spreadsheet Loader toolbar, click **Upload**.
Data is imported to the stage tables. Objects that import successfully are loaded automatically to the application tables.
2. Click **Refresh**.

Errors from both import and load are reported automatically in the **Progress** and **Message** columns of the Spreadsheet Line Status section of the spreadsheet.

3. Correct any errors in the spreadsheet.
4. Repeat from step 1 of Uploading Jobs and Correcting Errors until all rows are imported and loaded successfully.
5. Close the spreadsheet.

Related Topics

- [Setting Up the Desktop Integration for Excel: Procedure](#)

FAQs for HCM Spreadsheet Data Loader

Why can't I access HCM Spreadsheet Data Loader?

You may not have the correct permissions. You must have the Human Capital Management Integration Specialist job role or privileges.

In addition, the **HCM Data Loader Scope** configuration parameter must be set to **Full**.

How can I fix errors?

You fix errors in the spreadsheet from which you uploaded the data. Errors are reported in the **Progress** and **Message** columns of the Spreadsheet Line Status section of the spreadsheet when you click **Refresh**. After correcting any errors, click **Upload** again.

Can I load date-effective data?

Yes. You can create, update, and correct date-effective objects. To load date-effective history, you provide a separate row for each date range.

Can I change first and last effective dates when loading date-effective objects?

Yes. Include the **Replace First Effective Start Date** and **Replace Last Effective End Date** attributes in the spreadsheet template, as appropriate. Spreadsheet users set these attributes to **Y** and provide revised dates on the **Effective Start Date** and **Effective End Date** attributes.

For some objects, you can change only one of these dates.

Can I load translated objects?

Yes. You don't have to create a separate spreadsheet template for the translation object. Instead, include translation attributes, such as **Language**, in the spreadsheet template for the base object. You can use spreadsheets generated from the template to maintain both base-object and translation data.

18 Postload Processing

Processes to Run After Loading Data: Explained

After a successful data migration or incremental update, you must run processes to complete the setup of some business objects. Run these processes in the Scheduled Processes work area. Select **Navigator - Tools - Scheduled Processes**.

This table identifies processes to run after loading data and the object to which each process applies. For worker objects, the table also suggests the order in which to run the processes when you're running them all manually. By default, these two processes run automatically when a load of worker objects completes:

- Refresh Manager Hierarchy
- Update Person Search Keywords

You can prevent either or both of these processes from running automatically using a SET instruction in the Worker.dat file.

Process	Business Object	Order	Description
Synchronize Person Records	Worker	1	<p>Notifies consuming Oracle Cloud applications, such as Oracle Fusion Trading Community Model, of changes to person and assignment details since the last data load.</p> <p>Submit this process with the After Batch Load parameter set to Yes. In the From Date field enter the start date of the load. In the To Date field enter the end date of the load.</p>
Refresh Manager Hierarchy	Worker	2	<p>For performance reasons, the complete manager hierarchy for each person is extracted from active data tables. The hierarchy is stored in a separate manager hierarchy table, known as the denormalized manager hierarchy. This approach ensures that a person's manager hierarchy is both easily accessible and up to date. The Refresh Manager Hierarchy process populates the denormalized manager hierarchy table with latest information after each data load.</p>
Update Person Search Keywords	Worker	3	<p>Several attributes of person, employment, and profile records are used as person-search keywords. This process copies keyword values in all installed</p>

Process	Business Object	Order	Description
			<p>languages from the originating records to the PER_KEYWORDS table, where they're indexed to improve search performance.</p> <p>Enter a person name in the Name field to run the process for a single worker. If you leave the field blank, then the process runs for all workers, which can be time consuming. To avoid performance problems, submit the process with the After Batch Load parameter set to Yes. In this case, the process creates keywords for new records but doesn't update those for existing records.</p> <p>Leave the Batch ID field blank.</p>
Optimize Person Search Keywords Index	Worker	4	<p>Optimizes the index of the PER_KEYWORDS table to improve search performance. Run this process after the Update Person Search Keywords process completes. By default, the process runs for a maximum of 180 minutes and performs a full optimization. You can enter a different Maximum Optimization Time. If you select the Rebuild the Index option, then the Maximum Optimization Time value is ignored. Rebuilding the index can be time consuming.</p>
Send Pending LDAP Requests	Worker	5	<p>Sends user-account requests to your LDAP directory. Run this process only when you want user accounts to be either created or updated.</p>
Send Personal Data for Multiple Users to LDAP	Worker	6	<p>Ensures that personal data held in your LDAP directory matches that held by Oracle HCM Cloud.</p>
Autoprovision Roles for All Users	Worker	7	<p>Provisions and removes user roles based on current role-provisioning rules.</p>
Apply Name Formats to Person Names	Name Format		<p>Applies the Display Name, List Name, Full Name, and Order Name formats to denormalized person-name data when you add or update a name format. You can run the process for</p>

Process	Business Object	Order	Description
			either a single format or all formats in a single country.
ESS process for trees flattening	<ul style="list-style-type: none"> Department Tree Node Organization Tree Node 		Flattens tree data to improve query performance against the hierarchical data, especially for hierarchical queries such as roll-up queries.
ESS process for trees audit	<ul style="list-style-type: none"> Department Tree Node Organization Tree Node 		Audits tree-structure metadata to verify that it conforms to all rules and ensures data integrity. You can view audit details, view messages, and correct any validation errors that the audit detects.

Related Topics

- [Send Pending LDAP Requests: Explained](#)
- [Sending Personal Data to LDAP: Explained](#)
- [Autoprovisioning: Explained](#)
- [The SET and COMMENT Instructions: Explained](#)

Running Postload Processes: Points to Consider

This topic provides additional guidance for some postload processes. In particular, it identifies when the processes are required and when they're optional.


Send Pending LDAP Requests

You must always run Send Pending LDAP Requests after loading worker records. When you run Send Pending LDAP Requests:

- User accounts are created for new workers.
- Roles are provisioned to user accounts as specified by current role mappings.

If appropriate role mappings don't exist when you load new workers, then user accounts are created but no roles are provisioned. User accounts without roles are automatically suspended when Send Pending LDAP Requests completes. Also, subsequent role-provisioning requests to the suspended user accounts may be slow to process. To avoid this scenario:

1. Create a role mapping to provision at least one role, for example, Employee, to every current worker before you load worker records.
2. Prevent the creation of user accounts for terminated workers. Use the Manage Enterprise HCM Information task in the Setup and Maintenance work area to set the **User Account Creation for Terminated Workers** option to **No**.
3. Load worker records.

 **Note:** Always create a role mapping for the workers you're loading before you load them. Having this role-provisioning rule in place prevents new user accounts from being suspended unnecessarily.

Autoprovision Roles for All Users

You don't have to run this process after the initial or only load of worker records if you create role mappings to autoprovision roles before you load workers. However, if you load the workers before creating the role mapping, then you must run this process followed by Send Pending LDAP Requests for the loaded workers.

You don't have to run this process after an incremental update. However, you're recommended to run this process whenever a change is made to the autoprovisioning rules.

Send Personal Data for Multiple Users to LDAP

Run this process after the initial or only load of worker records. Run it after an incremental load:

- For changed users only
- If you're loading workers separately from their managers
- If updates were made to personal data by means of SQL scripts

Related Topics

- [Role Mappings: Explained](#)
- [Creating a Role Mapping: Procedure](#)

Validating Loaded Data: Explained

Once HCM Data Loader loads your data successfully, you're recommended to validate that the loaded objects appear as expected. This topic identifies some of the ways in which you can validate loaded data. These include use of:

- Manual testing
- Oracle Transactional Business Intelligence (OTBI) reports
- HCM Extracts

Using Manual Testing

You can:

- Validate data interactively by searching for a representative sample. For example, you can search for workers in the Directory or Manage Person work area. If a worker is also a manager, then you can explore the manager hierarchy.
- Validate work structures by searching for them. For example, you can search for jobs on the Manage Job page and locations on the Manage Locations page. Alternatively, you can test that they're available for selection. For example, you can update a worker's assignment to select a different job or location.

Using Oracle Transactional Business Intelligence Reports

You can use predefined OTBI reports to validate some uploaded data. To run these reports:

1. Select **Navigator - Tools - Reports and Analytics** to open the Reports and Analytics work area.
2. In the Contents pane of the work area, open the folder **Shared Folders - Human Capital Management**. Many subfolders of the **Human Capital Management** folder contain a **Transactional Analysis Samples** folder, which contains the predefined reports.

To list the predefined reports for an area, select the relevant **Transactional Analysis Samples** folder in the Contents pane and click the **Browse Catalog** icon. A list of predefined reports for the selected area, such as Workforce Management, appears in the work area. For example, you can run the Job Details report to list active jobs and the Location Details report to list active locations.

You can also design your own reports using Oracle BI Publisher. To open Oracle BI Publisher, select **Create - Report** in the Contents pane of the Reports and Analytics work area.

Using HCM Extracts

You can use HCM Extracts to extract data from Oracle HCM Cloud and deliver it in various formats. These formats include CSV, XML, Microsoft Excel, HTML, RTF, and PDF.

Maintaining the Stage Tables: Procedure

When you load large volumes of data into Oracle HCM Cloud, the HCM Data Loader stage tables can increase in size rapidly. You're recommended to delete processed data sets that you no longer need in the stage tables. How often you delete stage-table data depends on the volume and frequency of your data loads. During data migration, consider deleting every large data set as its processing finishes. However, deleting stage-table data isn't reversible. Don't delete stage-table data that you want to report on or for which you want to extract error information.


You can delete:

- An individual data set on the Import and Load Data page
- Multiple data sets on the Delete Stage Table Data page


Deleting a Single Data Set

Follow these steps to delete a single data set from the stage tables:

1. Select **Navigator - Data Exchange - Import and Load Data** to open the Import and Load Data page.
2. In the Search Results section of the Import and Load Data page, select the data set that you want to delete. Use the search criteria to restrict the data sets that appear in this section.
3. Click **Delete**.
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 stage-table data without also deleting the source file. However, 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** here has no effect.

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

 **Tip:** You can review the status of the deletion process in the Processes table on the Delete Stage Table Data page.

Deleting Multiple Data Sets

Follow these steps to delete stage-table data for multiple data sets or data sets that meet specific conditions:

1. Select **Navigator - Data Exchange - Delete Stage Table Data** to open the Delete Stage Table Data page.
2. Set the search criteria to identify the data sets you that you want to retrieve. By default, all data sets created by the current user and that have not been updated in the last month are listed.
3. Confirm that all data sets in the Search Results table are to be deleted and click **Schedule**.
4. To delete the source files for the data sets from the Oracle WebCenter Content server, set the **Delete Source File** parameter to **Yes**.
5. Click **Submit**.

You can review the status of the deletion process by refreshing the Processes table on the Delete Stage Table Data page.

HCM Data Loader Extracts: Explained

HCM Extracts provides two extracts for HCM Data Loader:

Extract Name	Description
HCM Data Loader Data Set Summary	Run this extract for a specified data set. It provides status values and record counts for both the data set and each business object in the data set. It reports all error messages and identifies all records affected by each error.
Compensation Changes	Run this extract for a specified compensation run to extract assignment, salary, and bonus changes made by the run. This extract is most useful if you're implementing a coexistence scenario.

The output from both extracts is in XML format. You can provide a BI Publisher layout to reformat the output. You can also define options to automate delivery of the output.

Running the HCM Data Loader Extracts

To run either of these extracts, you:

1. Copy the predefined extract.
2. Compile the formulas for your copy of the extract.
3. Configure your copy of the extract.
4. Submit the extract.

Copying an HCM Data Loader Extract: Procedure

HCM Extracts provides two predefined extracts for HCM Data Loader:

- HCM Data Loader Data Set Summary
- Compensation Changes

This topic explains how to copy a predefined extract and compile the formulas for the copied extract. You can follow these instructions for either the HCM Data Loader Data Set Summary extract or the Compensation Changes extract.

Copying the Extract

Follow these steps to copy an HCM Data Loader extract:

1. Select **Navigator - Data Exchange** to open the Data Exchange work area.
2. On the Tasks panel tab, select **Manage Extract Definitions**.
3. On the Manage HCM Extract Definitions page, search for the extract.
4. Select the extract in the search results and click **Copy**.
5. In the **Copy Extract Definition** dialog box, enter a name for your copy of the extract and click **OK**.
A confirmation message appears when the copy is complete.

Compiling the Formulas for the Copied Extract

Follow these steps to compile the formulas:

1. On the Manage HCM Extract Definitions page, search for your copy of the extract.
2. Click the extract name in the search results to display the extract definition on the Manage HCM Extract Definitions page.
3. Click **Switch Layout** to edit the extract definition.
4. On the Edit Extract Definition page, select the **Extract Execution Tree** folder in the left-hand pane to show the Extract Execution Tree page. All formulas have an error status.
5. Click **Compile All Formula** to compile the formulas.
6. Refresh the table periodically until all formulas are successfully compiled. A green check mark means success.

Configuring a Copied HCM Data Loader Extract: Procedure

HCM Extracts provides two predefined extracts for HCM Data Loader:

- HCM Data Loader Data Set Summary
- Compensation Changes

This topic explains how to configure an extract after copying it and compiling its formulas. Follow these steps:

1. Select **Navigator - Data Exchange** to open the Data Exchange work area.
2. On the Tasks panel tab, select **Refine Extracts** to open the Refine HCM Extracts page.
3. On the Refine HCM Extracts page, search for your copy of the extract.
4. In the search results, select your extract and click **Edit**.
5. Click the Parameters tab to review the parameter definitions.

6. Edit the **Display, **Sequence**, **Parameter Basis**, and **Basis Value** for each parameter.**

This table shows the parameters that you must edit and the required values for the HCM Data Loader Data Set Summary extract. You can ignore parameters not shown here.

Parameter	Display	Sequence	Parameter Basis	Basis Value
Process End Date	No	3	Context Binding	System Date
Process Start Date	No	9		
Content ID	Yes	12		
Data Set Name	Yes	13		
Process ID	Yes	11		
Show Messages	No	14	Constant Bind	Y
Show Stack Trace	No	15	Constant Bind	N

This table shows the parameters that you must edit and the required values for the Compensation Changes extract. You can ignore parameters not shown here.

Parameter	Display	Sequence	Parameter Basis	Basis Value
Process End Date	No	3	Context Binding	System Date
Process Start Date	No	9		
Extract Target	No	15	Constant Bind	HDL
Period Name	Yes	12		
Plan	Yes	11		
Process Id	Yes	13		
Run Id	Yes	14		

7. Save your changes.

You can now run your copy of the extract.

Refer to the following HCM Extracts documentation on My Oracle Support for guidance on modifying the extract output and specifying delivery options:

1. Oracle Fusion HCM Extracts Guide (document ID 1559127.1)

2. Oracle Fusion HCM Extracts: Delivery Options (document ID 1573099.1)

Related Topics

- [Oracle Fusion HCM Extracts Guide](#)
- [Oracle Fusion HCM Extracts: Delivery Options](#)

Submitting an HCM Data Loader Extract: Procedure

HCM Extracts provides two predefined extracts for HCM Data Loader:

- HCM Data Loader Data Set Summary
- Compensation Changes

This topic explains how to submit an HCM Data Loader extract after configuring it. Follow these steps:

1. Select **Navigator - Data Exchange** to open the Data Exchange work area.
2. On the Tasks panel tab, select **Submit Extracts** to open the Submit a Process or Report page.
3. Using the **Query by Example** fields, find your extract and click **Next**.

The Enter Parameters page opens.

4. To submit the HCM Data Loader Data Set Summary extract, enter any one of the **Content ID**, **Process ID**, and **Data Set Name** parameters. Leave the **Process Configuration Group** parameter blank. This table describes the parameter values, which you can find on the Import and Load Data page.

Parameter	Description
Content ID	The content ID associated with the data set. This ID was assigned when you uploaded your source file to the Oracle WebCenter Content server.
Process ID	The process ID of the process that created the data set. This ID was assigned when you submitted your source file for processing by HCM Data Loader.
Data Set Name	The data set name, which is also your source file name.

To submit the Compensation Changes extract, enter a flow name for this execution of the extract. To identify the compensation run to report on, enter either the **Plan** and **Period Name** or the **Process Id** or the **Run Id**. This table describes the parameter values.

Parameter	Description
Plan	The compensation plan name.
Period Name	The period name for the compensation run.
Process Id	The process ID for the compensation run.
Run Id	The run ID for the compensation run.

Parameter	Description
-----------	-------------

5. Click **Next** until you reach the Review page.
6. Click **Submit**.

A confirmation message appears when your extract is submitted successfully.

7. Click **OK and View Checklist** in the **Confirmation** dialog box to open the extract results.

HCM Data Loader Error Analysis Report

The HCM Data Loader Error Analysis Report is a diagnostic test. It provides technical information that can help to resolve data-loading errors. Run this diagnostic test when error messages don't supply enough information for you to identify a solution. You can send the test output to Oracle Support, if necessary.

To run the HCM Data Loader Error Analysis Report test, select **Settings and Actions - Run Diagnostic Tests**.

Diagnostic Test Parameters

Content ID

A unique reference to a single data set. You can find the content ID in the **Content ID** column of the Search Results table on the Import and Load Data page. You must specify either the content ID or the process ID to identify the data set.

Process ID

A unique reference to a single data set. You can find the process ID in the **Process ID** column of the Search Results table on the Import and Load Data page. You must specify either the content ID or the process ID to identify the data set.

Business Object Name

The name of a business object in the data set, such as Worker or Job. You can find the business object name in the **Business Object** column of the Details table on the Import and Load Data page. This parameter is optional, but you're recommended to supply it if your data set contains multiple business objects.

Detailed Mode

Set this parameter to **Y** to include the details of any reported errors in the output, including message stack trace for the records in error.

Diagnostic Test Results

The diagnostic test output is in user-readable HTML format. It provides:

- Status information
- Details of every process submitted to process your data set
- All errors raised for each business object in the data set

- Message stack trace and nonsensitive attribute values for the records in error, if you set the **Detailed Mode** parameter to **Y**


No sensitive data appears in the test results.

Running the HCM Data Loader Error Analysis Report: Procedure

The HCM Data Loader Error Analysis Report is a diagnostic test that provides technical information about data loads to help you resolve errors. To submit this diagnostic test, you must have access to the Diagnostic Dashboard.

Follow these steps:

1. Select **Settings and Actions - Run Diagnostic Tests** to open the Diagnostic Dashboard.
2. In the Search for Tests section of the Diagnostic Dashboard, enter **HCM Data Loader Error Analysis Report** in the **Test Name** field and click **Search**.
3. In the search results, select the check box next to the test name and click **Add to Run**. The test appears in the Choose Tests to Run and Supply Inputs section of the page.
4. To review the parameters, click the **Click to Supply or Edit Input Parameters** icon in the **Input Status** column.
5. In the **Input Parameters** dialog box, select parameters to include in this test run and supply values for the selected parameters.

 **Note:** Use the parameters to restrict the output from the test to those errors that need investigation. For example, don't run the test for an entire data set if the errors relate to a single business object.
6. Click **OK**.
7. If required, enter a name for the test run in the **Run Name** field. The name can help you to locate your results easily. If you leave the field blank, then a name is generated.
8. Click **Run** to run the test and click **OK** to close the confirmation dialog box.
9. In the Diagnostic Test Run Status section, click the **Display Latest Test Run Status Information** icon to see the results of this test.
10. Expand the test hierarchy for your run to see the results. Click the **Report** icon to open the report.

You can save the output and attach it to a service request if required.

19 Payroll Data Loading

Overview

You can load payroll data for initial migration or mass data entry using the payroll batch loader, predefined processes, and web services. You can also automate the regular import of time cards, absence entries, and benefit enrollments using predefined flows.

This overview outlines your options to meet the following data loading requirements:

- Loading setup data
- Exporting and importing setup data between test and production environments
- Ongoing data loading

Loading Setup Data

You can use HCM Data Loader during implementation to migrate HCM setup data, including element entries and salaries. However, for most payroll-related setup data and worker data, as shown in the following table, use the payroll batch loader. You can load the data from a spreadsheet or a file.

Payroll Setup Data	Worker Data
Balances in balance groups	Assigned payrolls
Elements	Bank details for personal payment methods
Formula globals	Element entries
Object groups	Initial balance values
Payroll definitions	
User-defined tables	

Use the Batch Loader task in the Payroll Administration, Data Exchange, or Data Exchange work area to perform the following actions:

1. Enter the data in a workbook format that's specific to the object type or create a batch from a file using a transformation formula.
2. Save your entries to staging tables.
3. Validate the data in staging tables.
4. Submit the Transfer Batch flow to load the data to the application tables. For initial balance values, use the Load Initial Balances flow instead.

Exporting and Importing Setup Data Between Environments

Typically, you initially migrate data to a test environment. After successful testing, you can move the data to your production environment using one of the following tasks:

- Use the Manage Configuration Packages task in the Setup and Maintenance work area to export and import a configuration package.
- Use the Create Batch for an Object process in the Payroll Administration work area to select specific objects, such as elements or formulas, to migrate.

Ongoing Data Loading

You can use the payroll batch loader and predefined batch processes to load element entries and other payroll data on an ongoing basis. You can automate the submission of the flows using the Flow Actions web service.

Specific flows exist for loading payroll-related data for payroll processing, such as the Load Absence Batches process, where you specify the interface type and XML file containing the data to load.

To import data from a third-party payroll provider, such as processed payroll data or payslips, you can use the HCM Data Loader.

Related Topics

- [Implementation Project Based Export and Import: Explained](#)
- [Migrating Objects Using the Payroll Batch Loader: Procedure](#)

Initial Balance Loading: Explained

Setting initial balances values is an essential task when you migrate payroll data from another application. First you load balance values into batch views, then submit the Load Initial Balances process from the Payroll Calculation work area. The process validates the information then loads the batch.

Balance Initialization Elements

For each initialized balance, you must create elements in the Balance Initialization classification and add them to the balance as balance feeds.


You can create up to three balance initialization elements for the following levels:

1. Assignment
2. Employment terms
3. Payroll relationship

Each balance initialization element must:

- Be nonrecurring and for balance adjustments only.
- Be capable of being processed in the payroll run.

- Have an input value to feed the balance and an input value for each context required by the balance.

 **Tip:** If you need to set initial values for a large number of balances you can define multiple input values for a single element with each input value feeding a different balance.


- Have an eligibility record associated with it, with an early effective start date.

Batch Views

To populate the batch views with the balance values for the initialization date, use the Payroll batch loader workbook, which you can download from the Payroll Administration, Data Exchange, or Checklist work area.

The views include:

- PAY_BALANCE_BATCH_HEADER
- PAY_BALANCE_BATCH_LINES

 **Note:** The PAY_BALANCE_BATCH_LINES view has a complex definition. You can't directly insert balance values into this view. You must use the batch loader workbook.

When you create the batch header and lines, consider the following points:

- Divide your employees into separate batches to limit the size of each batch.
- Within a batch, ensure that you include batch lines for every balance to be initialized for a person. You can't split lines for a person across multiple batches.
- The date you specify on the batch header applies to all lines unless you enter an override date at the line level. The date at line level must be on or before the date on the header.
- You can't initialize balances once you have run any payroll processes.
- Every line must include a value for payroll relationship number. If the line is initializing a terms-level balance, you must also enter a term number. If the line is initializing an assignment-level balance, you must also enter a term number and an assignment number.

Related Topics

- [Payroll Batch Loader Tasks: Explained](#)

Initial Balances: How They're Loaded

The Load Initial Balances process validates then processes the initial balance values you load into batch views. It creates balance adjustments to set the required values. To load initial balances, use the Submit a Process or Report task in the Payroll Calculation work area.

Settings That Affect Initial Balances

The data you load into the batch views determines which defined balances the process initializes and the values it uses. Typically, you group employees into batches to manage the initialization of their balances.

How the Balances Are Initialized

The Load Initial Balances process validates the existence of the entities referenced in the batch data, including:

- Balances
- Balance dimensions
- Tax reporting units
- Payrolls
- Payroll relationships
- Employment terms
- Assignments

It also checks that values are available for the contexts used by each balance dimension. It doesn't check, for example, that an employee is assigned to a specified organization. After successful validation, the process creates balance adjustments.

Batch Line Status	Description
V	Batch line is valid.
T	Batch line has been successfully processed.

Examples

The following table shows a simple three-line batch loaded on 18 June.

Defined Balance	Value
Total_ Earnings_PTD	100
Total_ Earnings_QTD	250
Total_ Earnings_YTD	500

For this batch, the process creates an adjustment on the first day of the time period relevant to each dimension, as shown in the following table:

Adjustment Date	Adjustment Value	Balances Adjusted
1 June	100	Total_ Earnings_PTD Total_ Earnings_QTD Total_ Earnings_YTD
1 April	150	Total_ Earnings_QTD

Adjustment Date	Adjustment Value	Balances Adjusted
		Total_ Earnings_YTD
1 Jan	250	Total_ Earnings_YTD

Balance Batch Header and Lines Views

The Load Initial Balances process uses the PAY_BALANCE_BATCH_HEADER and PAY_BALANCE_BATCH_LINES views to hold the data to initialize balance values.

You must load data into these views using the Payroll batch loader workbook or API in the PAY_BALANCE_BATCH_LINES_PKG PL/SQL package.

Create each batch line with a BATCH_LINE_STATUS of U (unprocessed) and leave the PAYROLL_REL_ACTION_ID column blank. The batch upload process updates these two columns.

Required Columns


 **Note:** You can view the full column listing by querying the view. For more information about views, see the Tables and Views for Oracle HCM Cloud guide.

The PAY_BALANCE_BATCH_LINES view requires the columns shown in the following table. Where there is both an ID column and a name column for the same entity, for example, PAYROLL_ASSIGNMENT_ID and ASSIGNMENT_NUMBER, you can populate either column. You must, however, populate at least one. If you leave the ID column blank, the batch upload process uses the name column value to derive the ID value.

Column	Comments
PAYROLL_RELATIONSHIP_ID + PAYROLL_RELATIONSHIP_NUMBER	Identify the payroll relationship for this balance value.
BALANCE_TYPE_ID + BALANCE_NAME	Identify the balance for this balance value.
BALANCE_DIMENSION_ID + DIMENSION_NAME	Identify the balance dimension for this balance value. DIMENSIONNAME should be populated with the country-specific dimension usage dimension name rather than the global DIMENSION_NAME held on PAY_BALANCE_DIMENSIONS.
VALUE	Identify the numerical value of the balance on the upload date.
UPLOAD_DATE	Identify the date of the balance value. This date must be on or before the upload date for the batch header.
PAYROLL_ID + PAYROLL_NAME	Identify the context required for evaluating a balance value, even though it may not be a context for the dimension.

Core Context

If the balance dimension expects a value for this context, you must populate the core contexts shown in the following table.


Column	Comment
PAYROLL_ TERM_ID + TERM_NUMBER	Identify the payroll terms for this balance value, if required. Where there is both an ID column and a number column for the same entity, you can populate either column, but you must populate at least one.
PAYROLL_ ASSIGNMENT_ID + ASSIGNMENT_ NUMBER	Identify the payroll assignment for this balance value, if required.
LEGAL_ EMPLOYER_ID + LEGAL_ EMPLOYER_NAME	
TAX_UNIT_ID + TAX_UNIT_NAME	
AREA1 + AREA2 + AREA3 + AREA4	Identify state and county code information.
 Note: In Oracle E-Business Suite these items were STATE_CODE, COUNTY_CODE and so on.	
THIRD_ PARTY_PAYEE_ID + THIRD_ PARTY_ PAYEE_NAME	
TIME_ DEFINITION_ID + TIME_ DEFINITION_ NAME	
RUN_TYPE_NAME + RUN_TYPE_ID	
ELEMENT_ ENTRY_ID	
BALANCE_DATE	
CALC_ BREAKDOWN_ID	Identify the ID associated with the payroll terms. If the calculation breakdown is: <ul style="list-style-type: none"> • The calculation card, enter the PAY_DIR_CARDS_F.DIR_CARD_ID. • The calculation component, enter the PAY_DIR_CARD_COMPONENTS_F.DIR_CARD_COMP_ID.

Legislative or User-defined Context Columns

There are six legislative or user-defined contexts that you must populate if the balance dimension expects a value for these contexts. For each context, there is a context ID, name, and value. For example:

- CONTEXT1_ID + CONTEXT1_NAME

- If the balance dimension uses CONTEXT1 populate CONTEXT1_VALUE

 **Note:** The values of the core contexts are the IDs themselves, while the values of the legislative contexts are separate from the ID of the context.

Populate CONTEXT[1-6]_NAME with the name of the context usage for that legislative or user-defined context. Populate CONTEXT[1-6]_VALUE with the actual value of that context.

Related Topics

- [Tables and Views for Oracle HCM Cloud](#)


Adjusting Multiple Balances: Procedure

Manage balance adjustments that impact multiple employees by processing adjustments in a single batch using the payroll batch loader. To import multiple balance adjustments at the same time in a single batch for an element, complete the following sequence of steps.

1. Use the Batch Loader task in the Payroll Administration work area.

 **Note:** Alternatively, you can use the Batch Loader task in the Data Exchange, or Checklist work area.

2. Enter data into integrated Microsoft Excel workbooks using the Adjust Balance task action. Data columns in the workbook vary depending on the legislative data group and the selected element.

 **Note:** The batch loader doesn't support balance adjustments to different payrolls or different inclusion rules within the same batch. Each of the following columns must have the same value: Effective Date, Payroll, Include in Costing Process, Include in Payment Balance. If you have different values for these columns, you must create a separate batch for each combination.

3. Save the data to staging tables.
4. Use the Transfer Batch task in the Payroll Administration work area to move data to the balance adjustment interface tables.
5. Use the Adjust Multiple Balances task in the Payroll Calculation work area to take the data from the interface tables and create balance adjustment element entries for each line within the batch.

The Adjust Multiple Balances process updates the element entry record for each person included in the batch.

6. Verify the balance results of the adjustments using the View Person Process Results task in the Payroll Calculation or Payroll Checklist work areas.

Related Topics

- [Payroll Batch Loader Workbook for Balance Adjustments](#)
- [Importing Data Using the Payroll Batch Loader: Explained](#)

Adjusting Balances: Critical Choices

Use balance adjustment in exceptional circumstances when you can't correct the source data by submitting the appropriate recalculation process. For example you might process an adjustment to correct an initialized balance or correct a balance included in statutory reports at the end of a quarter or year. You can also use the balance adjustment process in situations where, an employee moves at the end of the year and you don't receive a notification in time. You might adjust the employee's tax balance and refund the amount collected by mistake.

You can adjust the balance of an existing element for one person or for multiple persons in a single batch.

Before performing a balance adjustment, review the following factors:

- Regulated limits and related balances
- Balance adjustment for a single person
- Balance adjustment for multiple persons
- Element and input values to adjust
- When to cost adjustments

If you are processing a payroll, also consider:

- When to pay refunds
- Reporting an adjustment

Regulated Limits and Related Balances

Balance adjustments update the selected balance but not related balances, because you aren't recalculating the element and its balances. Balance adjustments also bypass system validation, which means that limits aren't verified. Before performing an adjustment, consider whether related balances require adjustment, and whether your adjustment exceeds statutory regulated limits. For example, raising the upper limit of one balance might require correcting the lower limit of a related balance and ensuring that they both meet statutory limits.

Balance Adjustments for a Single Person

Adjust the balance for a person, for example, when you load the person's balances initially, and discover the balance is incorrect for a tax balance dimensions. Two factors determine where you adjust the balance: whether you processed the element in the payroll run for the person, and whether you need to adjust the context of the balance dimension.

You initiate a balance adjustment from the Payroll Calculation work area. From here, you can:

- Use the Adjust Individual Balances task
- Select Adjust Balances from the Action menu on the View Person Process Results page, Balance Results section.

The following table contrasts the two tasks.

Adjustment Tasks	Can Adjust Elements Processed in the Run	Can Update Context
Adjust Individual Balances	Yes	Yes
You can also create new balance results for element not processed in a payroll run.		

Adjustment Tasks	Can Adjust Elements Processed in the Run	Can Update Context
Adjust Balances	Yes	No, the context is the one used for the payroll run.

Balance Adjustments for Multiple Persons

Upload multiple balance adjustments for an element using the Batch Loader task in the Payroll Administration work area. You enter data into integrated Microsoft Excel workbooks, save the data to staging tables, and transfer the data to the balance adjustment interface tables. You use the Adjust Multiple Balances task in the Payroll Calculation work area to take data from the interface tables and create the balance adjustment element entries for each line within the batch.

When you create the batch using the payroll batch loader, consider the following points:

- **Data to load**
The data you load into the batch, such as element and input values, determines which balances to adjust and which values are used to adjust the balances.
- **Data columns**
The data columns display the same attributes available for the element when you process the adjustment using the Adjust Individual Balances task.
- **Batch size**
As best practice, use smaller batches to manage the process. For example, divide the balance adjustments into separate batches by legislative data group and limit the size of each batch to 5000 lines.
- **Batch processing**
Batches are single-threaded. Streamline processing by entering multiple batches and processing in parallel

Element and Input Values to Adjust

The best practice is to adjust the balance of an existing element, instead of creating a new element for balance adjustment purposes. Creating a new element isn't recommended because you must create the balance feeds and input values, and ensure that you adjust all the related balances.

When you create an element using the Manage Elements task, you complete a template to create the element definition. For some element classifications, submitting the template automatically creates related elements, such as results elements.

You adjust the appropriate input values required to adjust the balance. For example, you might enter a different start date for a pension plan element, or geography values for an earnings element. If you adjust an element processed in a payroll period, input values that you skip use the values from the original entry.

Refer to the following table to determine which element and input value to use for a balance adjustment.

Primary Classification	Secondary Classifications	Template Category	Element to Use for Adjustments	Element Input Values to Use for Adjustments
Standard Earnings	All	Standard	Base element	Pay Value
Supplemental Earnings	All	Standard	Base element	Pay Value

Primary Classification	Secondary Classifications	Template Category	Element to Use for Adjustments	Element Input Values to Use for Adjustments
Direct Payments	All	Standard	Base element	Pay Value
Employee Tax Credits	All	Standard	Results	Pay Value, Accrued, Arrears, Not Taken
Employer Charges	All	Standard	Results	Pay Value, Accrued, Arrears, Not Taken
Employer Taxes	All	Standard	Results	Pay Value, Accrued, Arrears, Not Taken
Involuntary Deductions	All	Involuntary Deduction	Results	Deductions Calculated
Involuntary Deductions	All	Involuntary Deduction	Maintenance	Amount Accrued, Not Taken Amount, Arrears Amount
Involuntary Deductions	All	Involuntary Deduction	Processing Fee Results	Fee Calculated, Initial Fee Calculated
Involuntary Deductions	All	Involuntary Deduction	Organization Fee Results	Fee Calculated
Involuntary Deductions	All	Involuntary Deduction	Person Fee Results	Fee Calculated
Pre-Statutory Deductions	All	Standard	Results	Pay Value, Accrued, Arrears, Not Taken
Pre-Statutory Deductions	Pre-Statutory Deductions Pension Plan Pre-Statutory	Benefit	Employee Contributions Employer Contributions Additional Contributions	Contribution Amount
Taxable Benefits	All	Standard	Base element	Pay Value
Voluntary Deductions	All	Standard	Results	Pay Value, Accrued, Not Taken
Voluntary Deductions	Voluntary Deductions Pension Plan After Tax	Benefit	Employee Contributions Employer Contributions Additional Contributions	Contribution Amount

The display options for the element and balance depend on the date you specify and the employment level associated to the element. For example, the balances available on the Manage Balance Adjustment page display only balances processed in the payroll run for the selected level of the employment hierarchy.

When to Cost Adjustments

Consider costing adjustments if you calculate payroll. As a guideline, if you cost the original balance and the adjustment changes the amount or account number, you cost the adjustment. To cost the adjustment, select the option to include the adjustment in the costing process on the Costing and Payment details section on the Adjust Individual Balances page.

The following table shows two examples.

Adjustment	Cost the Adjustment	Explanation
An earnings element or pretax deduction	Yes	You cost the original balance and the adjusted balance affects quarterly wage and tax reporting requirements.
An incorrect value of an initialized balance	No	Typically the cost of the balance was entered correctly in the previous application.

If you process a balance adjustment for an earnings element included in a distribution group, such as an overtime element, you can specify an override cost account for the amount.

If you process a balance adjustment for an element that uses distributed costing, such as an employer tax, the adjusted amount is costed to a suspense account. This situation occurs because the costing process doesn't associate the adjusted balance of the distributed element with the earnings elements included in the distribution group. To correct the amount and costing for the balances of the earnings elements, process a cost adjustment for each earnings element.

For example, if the distribution group for an employer tax includes regular, overtime, and shift earnings elements, you would process a cost adjustment for the amount and costing of the regular, overtime, and shift earnings elements.

When to Pay Refunds


Some balance adjustments result in refunds that you process when calculating payroll. For example, when you adjust a balance, such as a tax that increases or decreases the amount owed, the balance is adjusted in the next payroll calculation for the person. If you adjust an input value that results in a refund, a payment is due to the person.

To pay the adjustment:

1. Select the option to include the adjustment in the payment balance on the Costing and Payment details section on the Adjust Individual Balances page.
2. Pay the refund by one of these methods:
 - o Process the payment in the next payroll run.
 - o Submit the Calculate Prepayments process to pick up the adjustment amount, and pay it as an external payment.

Reporting Balance Adjustments

Reports that you submit after you calculate the payroll include balance adjustments if the effective date of the balance adjustment falls in the date range you specify for the report submission parameters. For example, you might submit the Gross-to-Net report to view the results in the current payroll run. Or, you can submit the Payroll Activity to view balance adjustments before archiving year-end run results.

 **Note:** If you submit year-end reports, perform your balance adjustments before archiving the year-end payroll run results.

If you are using distributed costing, and you adjust the balance of an element included in a distribution group, such as an overtime earnings element, you can override the usual costing process by specifying a cost account for the amount.

If you adjust the balance of an element that uses distributed costing, such as an employer tax, the balance adjustment costing processes the amount to the suspense account. This situation occurs because no values for the earnings group are associated with the balance adjustment results. To correct the amount and costing of the distributed elements, process a cost adjustment for the results of each adjustment entry.

Related Topics

- [Payroll Batch Loader Workbook for Balance Adjustments](#)

Adjusting Balances: Worked Example

This example demonstrates how to adjust an individual balance for an employee whose initial balance was loaded incorrectly. The following table summarizes the key decisions for this scenario.

Decisions to Consider	In This Example
What effective date should I use for the adjustment?	June 30, 2013 This is the date the initial balance was loaded.
Do I adjust the base or results element?	Results element
Do I owe the person a refund?	No
Does the balance adjustment require costing?	No

Your enterprise initialized balances during a mid-year implementation on June 30. The initial balance included a life insurance deduction for Marie Kleim for 200 USD, instead of 300 USD. In the monthly payroll on July 31, Marie's payroll run results included a 50 USD deduction for her life insurance, which brings the year-to-date total to 250 USD. Marie received her July 31 payment, so it's not practical to roll back the run result to correct the underlying entry. Also, the balance adjustment to her life insurance deduction doesn't affect her taxable wages. You correct the balance by submitting an adjustment of 100 USD for Marie's life insurance deduction.

Adjusting an Individual Balance

1. From the Payroll Calculations work area, select **Adjust Individual Balances**
2. Search for and display Marie Kleim's record.
3. On the Manage Person Details page in the Adjust Individual Balances Element Details section, complete the fields as shown in this table. Use the default values except as indicated in this table.

Field	Value
Effective Date	June 30, 2013
Element Name	Life Insurance Results
Assignment Number	E33312345

4. In the Adjustment Values section, enter 100 USD for the Pay Value.

In this example, the results element for the pretax deduction uses the pay value as the calculated value.

5. In the Costing and Payment Details section, enter **No** for both questions.

You don't owe a refund, because the person's net pay doesn't change. You don't cost the adjustment, because the costing for the deduction was processed in the previous application before the mid-year implementation occurred.

Reviewing the Balance Adjustment

After completing the balance adjustment, you can review Marie's balance adjustment.

1. From the Payroll Calculation work area, select **View Person Process Results**.
2. On the View Person Process Results page, search for Marie Kleim's balance adjustment, processed July 31, 2013.
3. Click the link for Marie's record in the Search Results section.
4. On the View Person Process Results page, click the arrow to display the Balance Results section.
5. In the Search Balances section, complete the search criteria as shown in the table.

Field	Value
Search Criteria	Balance Group
Balance Group	Pretax Deductions for Statement of Earnings

The Search Results display the run result for the balance adjustment of 100 USD. When you later submit the August 31 monthly payroll run, the run results for Marie's new balance for the life insurance deduction display 400 USD, which includes the balance adjustment of 100 USD and the August 31 payroll deduction of 50 USD.

Related Topics

- [Viewing and Verifying Payroll Run Results: Points to Consider](#)
- [When do I cost a payroll balance adjustment?](#)

20 Payroll Batch Loader

Overview

Payroll managers and administrators use the batch loader for a variety of purposes when migrating and loading data during implementation and on an ongoing basis. This topic compares the different tasks to help you determine which approach to take when loading data.

You can use the batch loader tasks and processes to perform the following actions:

- Create, update, and delete data for supported objects
- Create elements and formulas using template questions
- Create batches from files
- Migrate objects between environments

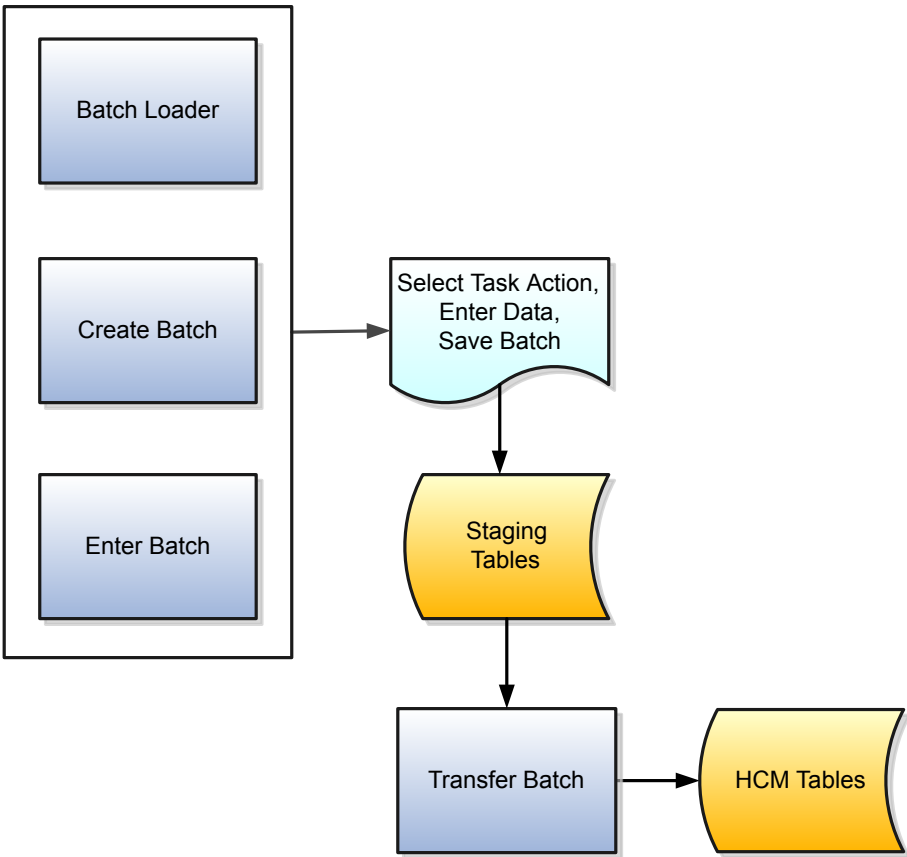
The following table describes the tasks and processes that you might use when working with the batch loader. You can access these tasks and processes in the Payroll Administration work area or using the Enter Batch task in a payroll flow.

Task or Process	Description	Comments
Batch Loader	Launches a workbook where you can create a batch or search for an existing batch.	Access this task in the Tasks pane.
Create Batch	Creates a batch to load element entries or balances by object group. Populates a workbook with all the elements or people in the object group you select when submitting the process.	Access this process from the Submit a Payroll Flow task.
Create Batch for an Object	Downloads some or all records and history for a specified object, such as an element or formula, into a batch for data migration.	Copy the batch lines to a new batch in another environment and submit the Transfer Batch flow to create the data. Access this process from the Submit a Process or Report task.
Enter Batch	For user-defined flows, launches a workbook for a single batch.	Sequence tasks in the flow pattern so that the Transfer Batch task follows the Enter Batch task. Access this process from the checklist of a user-defined flow pattern.
Load Batch from File	Uploads a file and transforms its data using the specified formula into a batch that you can transfer.	Access this process from the Submit a Payroll Flow task.
Purge Batch	Removes the specified batch.	You can purge a batch at any time. You can't transfer or view purged batches.

Task or Process	Description	Comments
		Access this process from the Submit a Process or Report task.
Transfer Batch	Creates entries in the applicable HCM tables for the specified batch.	Access this process from the Submit a Process or Report task.

Creating, Updating, and Deleting Data

The batch loader workbooks provide a fast way to upload batches of data. You create batches based on predefined templates, enter your data, load data into staging tables, and then transfer the batch into the HCM tables. The following figure illustrates the basic process and the tasks you use.




The task names in the workbook represent the type of object the batch loader supports. The task actions available depend on the task name you select, shown in the following table.

Task Name	Task Action
Balance	Adjust Balance

Task Name	Task Action
Balance Group	Add Balance Definition
	Remove Balance Definition
Bank and Branch	Create Bank
	Inactivate Bank
	Create Bank Branch
	Inactivate Bank Branch
Costing Setups	Create Cost Allocation
	Delete Cost Allocation
Element	Create Element Eligibility
	Delete Element Eligibility
	Create Input Value
	Delete Input Value
	Update Input Value
	Update Element
	Delete Element
	Create People Group Key Flexfield Segment Combination
	Delete People Group Key Flexfield Segment Combination
Element Entry	Create Element Entry
	Update Element Entry
	Delete Element Entry
External Bank Account	Add Person to Existing Account
	Create External Bank Account
	Inactivate External Bank Account
Fast Formula	Create
	Update
	Delete
Fast Formula Global	Create
HCM User-Defined Table	Create User-Defined Table

Task Name	Task Action
	Update User-Defined Table
	Delete User-Defined Table
	Create User-Defined Column
	Update User-Defined Column
	Delete User-Defined Column
	Create User-Defined Row
	Update User-Defined Row
	Delete User-Defined Row
	Create User-Defined Column Instance
	Update User-Defined Column Instance
	Delete User-Defined Column Instance
Payment Method	Create Personal Payment Method
	End-Date Personal Payment Method
	Delete Personal Payment Method
	Create Third-Party Personal Payment Method
	Delete Third-Party Personal Payment Method
	Update Third-Party Personal Payment Method
	Create Third-Party Organization Payment Method
	Delete Third-Party Organization Payment Method
	Update Third-Party Organization Payment Method
Payroll Definition	Create Payroll Definition
	Update Payroll Definition
	Delete Payroll Definition
Payroll Relationship	Add Payroll
	Delete Payroll
	End Payroll
	Update Assigned Payroll Details
	Update Assignment Details
	Update Payroll Element Duration Date
Person	Person Delivery Method

Task Name	Task Action
	Person EFF
	Person Legislative Data
Third-Party Organization	Create Third-Party Organization
	Update Third-Party Organization
	Merge Address

 **Note:** You can have only one task action per workbook. If you require multiple task actions for a large set of data to load, use the Load Batch from File process.


Related Topics

- [Creating Globals Using the Batch Loader: Worked Example](#)
- [Creating Element Entries Using the Batch Loader: Worked Example](#)
- [Loading a Payroll Batch From a File: Procedure](#)

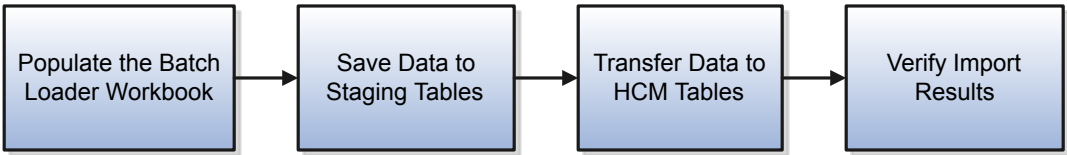
Importing Data: Procedure

Use the Batch Loader task in the Payroll Administration, Data Exchange, or Checklist work area to import data from integrated Microsoft Excel workbook templates into the staging tables and transfers data into the application.

The upload tasks available vary depending on the data type that you want to load. Each workbook template includes required and optional columns for loading data.

 **Tip:** You perform tasks with interdependencies sequentially, creating separate workbooks for each task. For example, a bank must exist before you associate its branches, so you first create banks in one workbook, and then create the branches in the next workbook.

The following figure illustrates the basic process for importing data using the batch loader.



Data Exchange Work Area, Payroll Administration, and Checklist Work Areas

In the Data Exchange, Payroll Administration, and Checklist work areas, you can:

- Create, update, and delete data for supported objects.

- Create elements and formulas using template questions.
- Create batches from files.
- Transfer the batches into the HCM tables.
- Verify the imported results.

Additionally, in the Data Exchange work area, you can run archives and extracts and view the results.

For Payroll Batch Loader, the Data Exchange work area offers user friendly options to:

- View the batches by status, such as Completed and Requires Attention.
- Download the log file, correct the erroneous processes, and resubmit them.

For Extracts, the Data Exchange work area offers options to:

- Submit an Extract Process to run immediately or at a later time.
- Monitor the process runs, including the number of records archived and extracted, and errors at each stage.
- View the process results by status.
- Download the log file, correct the erroneous processes, and resubmit them.

Use the Payroll Administration and Checklist work areas provide access to detailed checklists for tasks that have child tasks. For example, a custom process that has several batch loader tasks with in a flow.

Related Topics

- [Creating Element Entries Using the Batch Loader: Worked Example](#)
- [Payroll Batch Loader Workbook for Payroll Definitions](#)
- [Payroll Batch Loader Workbook for Payroll Relationships](#)
- [Creating Globals Using the Batch Loader: Worked Example](#)

Creating a Report to View Batch Line Errors: Worked Example

This example describes how to create a flow to view a report showing errors that occurred while transferring a batch using the payroll batch loader. After you create the report and the flow pattern for it, you can add it to other flow patterns. For example, for a custom flow pattern that includes the Transfer Batch flow task, you could add this flow immediately after that task.

In this example, the SQL query that defines the data model for the report takes the batch name as a bind variable. The bind variable enables the batch name to be user-entered when run alone, or dynamically derived when run within another flow.

Summary of Tasks

This worked example includes details for the following tasks:


1. Creating the report data model and output file name

2. Creating the report template
3. Creating the flow pattern to submit the report


Prerequisites

This worked example assumes you completed the following prerequisites:

1. The following software is installed:
 - o JRE version 7 or later
 - o Microsoft Word
 - o Oracle BI Desktop Integration for Word

 **Tip:** Select the Download BI Desktop Tools menu in the Get Started section of the Oracle Business Intelligence home page to download the Template Builder for Word installer.


2. You know the name of an existing batch that has transferred with errors. You will use this batch name for testing your report.

 **Tip:** You can use the following SQL query to retrieve the batch names: `select * from pay_batch_headers where batch_status = 'E';`

Creating the Report Data Model and Output File Name


1. Sign in as a user with the Oracle Business Intelligence administration privileges.
2. Create a SQL-based data model for the report as follows:
 - a. In the Navigator, select **Reports and Analytics**.
 - b. On the Reports and Analytics work area, click **Browse Catalog**.
 - c. On the Catalog page, select **New**, and then select **Data Model**.
 - d. On the Diagram tab, click **New Data Set**, and then click **SQL Query**.
 - e. In the SQL Query window, enter values as shown in this table, and then click **OK**.

Field	Value
Name	Batch Lines Report
Data Source	ApplicationDB_ HCM
Type of SQL	Standard SQL
SQL Query	select 'Marker' as Marker, bl.batch_line_id bl.batch_line_status line_text, bl.line_sequence from pay_batch_headers bh, pay_batch_lines bl, pay_message_lines ml where bh.batch_id = bl.batch_id and bl.batch_line_id = ml.source_id and bh.batch_name = :batchName

 **Note:** Ensure that you don't include a semicolon at the end of your SQL query.


- f. In the Add Parameter window, select **batchName**, and then click **OK**.
- g. On the Parameters page, in the **Display Label** field, enter **Batch Name**.
3. To change the name of the output file from Default Document.pdf, follow these steps:
 - a. In the left pane, select **Bursting**.
 - b. On the Bursting page, click **Create New Bursting**.
 - c. Set field values as shown in this table.

Field	Value
Data Source	ApplicationDB_ HCM
Split By	/DATA_DS/G_1/MARKER
Delivery By	/DATA_DS/G_1/MARKER
SQL Query	select 'Marker' as KEY, 'Batch Lines Report' as TEMPLATE, 'PDF' as OUTPUT_FORMAT, 'Batch Lines Report - ' to_char(sysdate 'YYYY-MM-DD: HH:MI:SS') as output_name from dual

 **Note:** Ensure that you don't include a semicolon at the end of your SQL query.

4. Click the **Save As** icon and save the data model under the path **Shared Folders/Custom/** with the name **BatchReportDataModel**.
5. Test that the data model query returns data as follows:
 - a. In the left pane, click **Data Sets**.
 - b. On the Data tab, enter the name of an existing batch that transferred successfully with errors, and then click **View**.

You should see values for BATCH_LINE_ID and BATCH_LINE_STATUS and any messages that occurred.


 **Note:** The batch name field is case sensitive, so enter the name exactly. If you don't see values or if you entered incorrect SQL syntax, edit the SQL for the data model and try again.

6. Click **Save As Sample Data**, and then click **OK**.
7. Click **Export**, and then save the XML file to your file system. You will use this when designing your report template.
8. Click **Save**.

Creating the Report Template

1. Generate the report template as follows:
 - a. On your data model page, click **Create Report**.
 - b. In the Create Report window, click **Cancel**, and then click **OK**.
 - c. Click the **Select Data Model** icon, select your data model, and then click **OK**.
 - d. Click the **Generate** icon.
 - e. In the **Template Name** field, **Batch Lines Report**, and then click the **Generate** button.
 - f. Click **Edit** and open the file in Microsoft Word.

- a. Save the RTF file to your file system.
2. In Microsoft Word, modify your template as follows:
 - a. Delete the **MARKER** and **BATCH_LINE_ID** columns from the table.
 - b. Change the column headings to user-friendly values, such as Status, Error Text, and Line Number.
 - c. Click the **Field** icon and insert **BATCHNAME** before the batch lines table.
 - d. Insert text before the BATCHNAME field, such as **Batch Lines Errors for Batch** followed by a colon and space character.
 - e. Make any other layout changes, as needed, such as changing column width.
 - f. Save and close the RTF file.
3. Upload your modified template as follows:
 - a. On the report page, under your report icon, click **Properties**.
 - b. Click the **Upload** icon.
 - c. Click **Browse** to select your RTF template, and then click **OK**.
 - d. In the Upload Template File window, in the **Locale** field, select the same local that is listed as the default local, and then click **OK**.
 - e. Click **OK** to overwrite the existing template for that locale.
 - f. Click **Save**, and then click **Return**.
4. Set the default format as follows:
 - a. At the top-right corner, click **View a List**.
 - b. In the **Default Format** field, select **PDF**.
 - c. Click **Save Report**.
 - d. In the Save As window, in the **Name** field, enter **Batch Lines Report**, and then click **OK**.
5. Ensure that the report has the correct role access as follows:
 - a. On the Catalog page, locate the batch lines report.
 - b. On the More menu for Batch Lines Report, click **Permissions**.
 - c. Click **Add users/roles** to add roles, as necessary.

 **Tip:** If you have several roles to add, search for and add the BI Consumer role with full control, which most other roles inherit.

- d. Click **OK**.
6. Test your report as follows:
 - a. On the Catalog page, for the batch lines report, click **Edit**.
 - b. Ensure that the name of your data model displays above the report icon. If it doesn't, add it and save your changes.
 - c. On the Catalog page, for the batch lines report, click **Open**.
 - d. In the **Batch Name** field, enter the batch name you entered previously. You should see the data in the sample report.

Creating the Flow Pattern

1. Ensure that you are signed in as a user with payroll administration privileges.
2. On the Payroll Checklist work area, select the **Manage Payroll Flow Patterns** task.
3. Click **Create**, and then click **Continue**, with no legislative data group selected.
4. On the Basic Information page, enter the values as shown in this table, and then click **Next**.

Field	Value
Flow Pattern	Batch Lines Report
Description	Submit a report to view batch line errors for a specified batch transferred using the payroll batch loader.
LDG Required	Optional
Default Flow	Yes
Activities to Include	Statutory
Selected Tasks	Run BI Publisher Report

5. Click **Next** to accept default values until you are on the Parameters page.
6. Add the Batch Name parameter as follows:
 - a. Click **Select and Add**.
 - b. Search for and select **First Argument**, and then click **OK**.
 - c. Click **Edit** and change its values as shown in this table, and then click **Next**.

Field	Value
Flow Parameter	Batch Name
Use for Searches	No
Display	Mandatory
Display Format	Text
Sequence	2
Usage	Input parameter
Parameter Basis	Post SQL Bind
Basis Value	select batch_name from pay_batch_headers where batch_id = :BATCH

7. On the Task Parameters page, edit the **Report Path** parameter as shown in this table.

Field	Value
Display	Yes
Parameter Basis	Constant Bind

Field	Value
Basis Value	/Custom/Batch Lines Report.xdo

8. Select the **First Argument** parameter and ensure that its values are as shown in this table.

Field	Value
Parameter Basis	Bind to Flow
Basis Value	Batch Name

9. Click **Next**, and then click **Submit**.
10. Test your flow as follows:
- In the Payroll Checklist work area, select the **Submit a Process or Report** task.
 - Select **Batch Lines Report**, and then click **Next**.
 - Enter a unique name for the current flow instance.
 - Enter the same batch name that you used when creating the report, and then click **Submit**.
 - Click **OK and View Checklist**, and then click the **Refresh** icon until the status shows as completed.
 - In the row for Run BI Publisher Report, click **Go to Task**.
 - On the Process and Reports tab, click **View Results**.
 - Click the PDF file name to open the report.

Related Topics

- [Creating a Flow within a Flow: Worked Example](#)

Task Actions for Migrating Elements: Explained

The task action you select to migrate elements in the payroll batch loader workbook determines the columns in the workbook. You must select the correct task action so that the workbook columns contain the appropriate element template questions for your country extension and element classification. Use the same task action in the batch to download the original element and in the batch to upload that element to another environment.

This topic describes which task actions to select for each of the following country extensions:

- Human Resources or None
- Payroll Interface
- Payroll

Check your country extension on the Manage Features by Country or Territory page. Refer to the following sections to ensure that you select the appropriate task action.

Human Resources or None

For all element classifications, use the **Quick Create - Basic Element Template** task action for implementations that don't process payroll information

Payroll Interface

The following table lists the two available task actions to select for elements created for implementations that share payroll-related data with third-party payroll providers.

Primary Classification	Task Action Name
Standard Earnings	Global Earnings Basic
All other classifications	Basic Element Template

 **Note:** All task action names begin with the prefix, **Quick Create -** .

Payroll

The following table lists the two available task actions to select for elements created for implementations that share payroll-related data with third-party payroll providers.

Primary Classification Base Name (and Display Name Variations)	Secondary Classification	Task Action Name
Absences		Global Absences
Balance Initialization		Basic Element Template
Direct Payments (Nonpayroll Payments)		Global Earnings
Employee Tax Credits		Global Tax Credits and Charges
Employer Charges (Employer Liabilities)		401K Deduction Template Global Tax Credits and Charges
Employer Taxes		Global Tax Credits and Charges
Information		Basic Element Template
Involuntary Deductions	Alimony Child Support Spousal Support	Support

Primary Classification Base Name (and Display Name Variations)	Secondary Classification	Task Action Name
	Bankruptcy Order	Court Order
	Creditor Debt or Wage Attachment	
	Court Order or Garnishment	
	Debt Collection	
	Educational Loan	
	Employee Requested	
	Regional Tax Levy	
	Tax Levy	Tax Levy
Pre-Statutory Deductions (Pretax Nonstatutory Deductions)	Deferred Compensation Plan	401K Deduction Template
	Pension Plan Pre-Statutory	Global Pension Pre-Statutory
	All others	Global Pre-Statutory Deduction
Social Insurance Deductions (National Insurance Deductions, Social Security Deductions, Superannuation Deductions)		Basic Element Template
Standard Earnings (Regular Earnings, Ordinary Wages)	Regular	Global Earnings Global Time Card
	Premium	Global Earnings Premium
Supplemental Earnings (Irregular Earnings, Allowances, Additional Wages)		Global Earnings
Tax Deductions (Employee Tax Deductions, PAYE Deductions)		Basic Element Template
Taxable Benefits (Imputed Earnings, Benefits in Kind)		Global Earnings
Voluntary Deductions	Pension Plan After Tax	Global Pension After Tax
	All others	Global Voluntary Deduction

 **Note:** All task action names begin with the prefix, **Quick Create** - .


Related Topics

- [Selecting Country Extensions: Critical Choices](#)

Migrating Objects: Procedure

Using the Create Batch for an Object process in the Payroll Administration work area, you can migrate objects between environments. To copy or migrate elements or formulas, follow these steps:

1. In the source environment, submit the **Create Batch for an Object** process in the Payroll Administration work area.
2. Enter a name for the flow to create the batch.
3. Enter a name of the batch to create and select a process date.
4. In the **Object** field, enter either **Element** or **Fast Formula**.
5. In the **Record** field, enter all or part of the object name to narrow results.

 **Tip:** Use the percent sign (%) as a wildcard character for partial names.

6. In the **Legislative Data Group Level** field, search for and select **Yes**.
7. In the **Process Configuration Group** field, search for and select the name of the group to use to control logging preferences.
8. Download the batch loader workbook and search for and double-click the batch you created.
9. On the Batch Content sheet, select the task action that is appropriate for your object. Double-click the task action to populate the columns with your object's attribute values.
10. In the target environment, use the Batch Loader task in the Payroll Administration work area to create a batch.
11. On the Batch Content sheet, search for and select the same task action as the original batch. Double-click the task action to display the columns.
12. Copy and paste the corresponding lines from the source environment's workbook to the target environment's workbook.
13. Save and transfer the batch on the target environment.

FAQs for Payroll Batch Loader

How many records can I load using the payroll batch loader?

You can load as many as 500,000 record in a single batch. If you require loading more than 500,000 records, split them up into separate batches.

How do I cancel a batch loader task?

You can use the Roll Back Batch Transfer process to roll back a batch transfer process only if it used the Create Element Entry task name in the payroll batch loader workbook. You can cancel the CreateBatchFromFile process and the Transfer HCM Upload Entry Batch process on the Scheduled Processes work area.

Why can't I load bank account numbers for personal payment methods?

The Transfer Batch process for bank account numbers will fail if any masking types are enabled in the environment. Ensure that the masking type for bank account numbers on the Manage System Security Options page is set to display all digits and retry the Transfer Batch process.

21 Payroll Batch Loader Using Workbooks

Payroll Batch Statuses: Explained


The payroll batch loader workbooks display a status on the Batch Header Sheet. The status reflects the status of the batch header, all the batch lines, and any control totals specified for the batch. Use the Batch Loader task from the Payroll Administration, Data Exchange, or Checklist work area to download the workbooks.

On the Batch Header Sheet, you can see the following status values.

Status	Meaning
Valid	All of the lines, totals, and header are valid
Transferred	All of the lines, totals, and header are transferred from the staging tables to the applicable HCM tables
Transfer incomplete	The header and totals are transferred along with only some of the lines
Unprocessed	At least one line, total, or the header is unprocessed, and no lines transferred
Error	The header wasn't transferred and at least one line, total, or the header is in error

Creating Element Entries Using the Batch Loader: Worked Example

This example demonstrates how to create element entries for bonus earnings for two workers in the US Sun Power legislative data group using the payroll batch loader. You select the annual bonus element and create element entries in the workbook entering the monetary amount for each bonus payment.

 **Note:** In this example, you create batch header and batch content and transfer the batch from the Data Exchange work area. You can also create and transfer batch from the Payroll Administration or Checklist work area.

Prerequisites

This worked example assumes that you completed the following prerequisites:

1. Install Oracle ADF Desktop Integration Runtime Add-in for Excel
2. Create a bonus element at the assignment level for element entries

3. Identify the assignment numbers of the workers who receive the bonus element entries


Creating a Batch Header

1. From the Data Exchange work area, select the Batch Loader task.
 2. On the Manage Batch Loader Processes page, click **Download Spreadsheet**.
 3. After the DesktopGenericBatchLoader.xlsx file downloads, open the file.
 4. When prompted to connect, click **Yes**.
 5. In the Login dialog box, enter your user ID and password, and then click **Sign In**.
 6. In the Batch Name column of the Search Results section, enter US Sun Power Bonus.
 7. In the Legislative Data Group list, select **US Sun Power**.
 8. Click **Save**.
 9. In the Upload Options dialog box, accept the default selection and click **OK**.
- Once your selections are saved, the batch is created and the status for that row displays that the row inserted successfully.

Creating Batch Content

1. On the Batch Header Sheet, double-click the batch name **US Sun Power Bonus** to prepare for data entry.
2. Navigate to the Batch Content Sheet.
3. Under Batch Contents Action, click **Add**.
4. Search for and select the task and task action as follows:
 - a. In the Task Name field, enter **Element Entry**.
 - b. In the Task Action Name field, enter **Create**.
 - c. In the Reference field, enter the element name for the bonus. For example, US Annual Bonus.
 - d. Click **Search**.
 - e. Select the row with your bonus element in the Reference column, and then click **OK**.
5. Double-click the **Create Element Entry** task action name.
The workbook refreshes to display the columns for the selected element, ready for data entry.
6. In the Batch Content Line Details section, enter the values for Nancy as shown in this table.

Field	Value
Line Sequence	1
Effective Start Date	2014-12-15
Assignment Number	E1000842
Pay Value	500

 **Note:** One number to identify the person is required. Use one of the number columns depending on the person's employment level. For example, if a person has more than one assignment for a single payroll relationship, enter both the payroll relationship number and the assignment number.

7. Right-click the next row number and insert a row for Joseph's details.
8. Enter the values for Joseph as shown in this table.

Field	Value
Line Sequence	2
Effective Start Date	2014-12-15
Assignment Number	E1003564
Pay Value	400

9. Click **Save**.
10. In the Upload Options dialog box accept the default selection and click **OK**.
Keep the workbook open. You verify the transfer of the element entries in the final step of this example.

Transferring the Batch

1. On the Batch Loader Processes page, click **Submit a Process**.
2. In the Legislative Data Group field, select **US Sun Power**.
3. In the Batch Process field, search for and select **Transfer Batch**.
4. On the Submit a Batch Process: Transfer Batch page, in the Batch Run Name field, enter a name for the batch, such as US Sun Power Bonus Batch.
5. In the Batch field, search for and select US Sun Power Bonus.
6. Click **Submit**.
7. Click **Refresh** until the Transfer Batch process status displays as complete.

Monitoring and Verifying the Transfer

On the Manage Batch Loader Processes page, you can click:

- **Recently Completed** to view the list of recently completed batch processes.
 - **Attention Required** to view the list of errored processes.
1. Click **Recently Completed**.
 2. In the Search field, search for and select the US Sun Power Bonus batch name to view the progress of the batch process.
On this page, you can monitor the progress of existing batch processes, view the volume of batch records processed, and troubleshoot any errored processes.
 3. You should see the status of the batch process as Complete.

Verifying the Transfer in the Workbook

1. Navigate to the Batch Header Sheet and double-click your batch name.
2. Navigate to the Batch Content Sheet.

Notice that the status displays as transferred.

3. Double-click the Create Element Entry task action name.

Notice that the two rows show that they are transferred.

4. Scroll to the right to show the Message column.


You should see no error messages. Nancy and Joseph should now have the bonus element entries. You can use the Manage Element Entries task to find the workers and view the element entries.

Related Topics

- Using Desktop Integrated Excel Workbooks: Points to Consider
- Payroll Batch Loader Tasks: Explained
- Payroll Batch Loader Workbook for Element Entries

Creating Globals Using the Batch Loader: Worked Example

This example demonstrates how to enter globals for two types of bonuses in the InFusion US legislative data group using the payroll batch loader workbook. You set the bonus values for executives at ten percent of their salary and instructors at a fixed value of 500. You can change these values later in the global to apply the same value in any formulas that use them.

 **Note:** In this example, you create batch header and batch content and transfer the batch from the Payroll Administration work area. You can also create and transfer batch from the Data Exchange or Checklist work area.

The following table summarizes key decisions for this scenario.

Decisions to Consider	In This Example
Load what type of data?	Fast Formula Global
Create which globals?	Executive Bonus for a percentage and Instructor Bonus for a fixed amount.

Prerequisite

This worked example assumes that you installed Oracle ADF Desktop Integration Runtime Add-in for Excel.

Creating a Batch Header

- From the Payroll Administration work area, select the Batch Loader task.
- On the Batch Loader page, click **Download**.

3. After the DesktopGenericBatch.xlsx file downloads, open the file.
4. When prompted to connect, click **Yes**.
5. In the Login dialog box, enter your user ID and password, and then click **Sign In**.
6. Navigate to the Batch Header Sheet at the bottom of the workbook.
7. In the Batch Name column of the Search Results section, enter InFusion Globals.
8. In the Legislative Data Group list, select **InFusion US**.
9. Click **Save**.
10. In the Upload Options dialog box, accept the default selection and click **OK**.

After you save your selections, the Batch Status text for that row displays that the row inserted successfully.


Creating Batch Content

1. On the Batch Header Sheet, double-click the batch name **InFusion Globals** to prepare for data entry.
2. Navigate to the Batch Content Sheet.
3. Under Batch Contents Action, click **Add**.
4. In the Task Name field, enter Fast Formula Global.
5. Click **Search**.
6. Select **Fast Formula Global**, and then click **OK**.

The workbook should update to display the columns for the selected task, ready for data entry.

7. In the Batch Line Content Details section enter the values for each global as shown in this table.

Field	First Global Value	Second Global Value
Line Sequence	1	2
Effective Start Date	2011-01-01	2011-01-01
Effective End Date	2020-12-31	2020-12-31
Value	.10	500
Data Type	N	N
Name	Executive Bonus	Instructor Bonus

 **Tip:** To insert more lines for additional global values, right-click on the row number where you want to add a row, and then select **Insert**.

8. Click **Save**.
9. In the Upload Options dialog box accept the default selection and click **OK**.

Keep the workbook open. You verify the transferred globals in the final step of this example.

Transferring the Batch

1. From the Payroll Administration work area, click **Submit a Process or Report**.
2. In the Legislative Data Group field, select **InFusion US**.
3. In the Flow Pattern column, select **Transfer Batch**, and then click **Next**.
4. In the Payroll Flow field, enter a name for the process, such as InFusion Globals Batch.
5. In the Batch field, search for and select **InFusion Globals**, and then click **Next**.
6. On the Enter Flow Interaction page, click **Next**.
7. On the Review page, click **Submit**.
8. Click **Refresh** until the Transfer Batch process status displays as complete.

Verifying the Transfer

1. In the workbook, navigate to the Batch Content Sheet.
You should see the status displayed as transferred.
2. Navigate back to the Batch Message Sheet.
You should see no error messages. You can use the new globals in your formulas.

Related Topics

- [Using Desktop Integrated Excel Workbooks: Points to Consider](#)
- [Payroll Batch Loader Tasks: Explained](#)
- [Loading a Payroll Batch From a File: Procedure](#)

FAQs for Payroll Batch Loader Using Workbooks

How can I access the payroll batch loader?

For payroll managers and administrators, select the **Batch Loader** task in the Payroll Administration, Data Exchange, or Checklist work area. If a flow includes the batch loader, you can also access it from the Payroll Checklist work area using the Enter Batch task on the Payroll Flow page.

Can I copy rows from another spreadsheet into a batch loader workbook?

Yes, but you must ensure that you save the workbook without errors before you transfer the batch. It's best practice to insert blank rows and then add content. If you have multiple rows to insert, copy rows from your offline spreadsheet, and then right-

click a row in the content area and select **Insert Copied Cells**. Once you verify that all the inserted data is in the correct columns, click **Save**. Then check that the status for each row shows that it was inserted successfully.

Can I use the batch loader to upload a spreadsheet I create?

No, you can't if you didn't download the payroll batch loader workbook. The batch loader workbook automatically inserts macros essential to successful processing. You can download the batch upload workbook to your desktop and edit the data before loading it.


How do I modify an Excel workbook template for the payroll batch loader?

You can't modify the structure of integrated Microsoft Excel workbooks because they're based on predefined templates. This restriction ensures the fields entered correspond exactly to the HCM tables that receive the uploaded data.

22 Payroll Batch Loader File Loading

Overview

Load Batch from File is a flow pattern that you can use to transform data in a source file into a payroll batch for upload. You can submit this flow alone or include it in a flow that you create for automating data loads on a periodic basis. When you submit the flow, either manually or using a web service, you can specify a transformation formula to transform the data, as needed.

 **Note:** You can create a payroll batch from a file from the Data Exchange, Payroll Administration or Checklist work area.

Assumptions

This procedure has the following assumptions:

- The user performing these steps has the Human Capital Management Integration Specialist role.
- You have the source file ready to upload to Oracle WebCenter Content, or it's already uploaded and you have noted the content ID. The following table shows three examples of how to upload files to the content server.

Example	Details
Upload File flow task	Browse to the source file on your file system and upload it as part of the Load Batch from a File flow. Using the Upload File flow task, you don't have to provide the file's content ID when you submit the flow.
UploadTool Java utility	Programmatically upload a file from a local or remote file system into the content server and retrieve its content ID. For more information, see the Automating HCM Data Loader attachment for the Using Payroll Batch Loader Guide (2030268. 1) on My Oracle Support at https://support.oracle.com .
Oracle WebCenter Content application	Browse to the source file on your file system, check it in to the content server, and retrieve its content ID. For more information, see Oracle Fusion Middleware Using Oracle WebCenter Content guide.

Prerequisites

Before you create a payroll batch from a file, ensure that you meet the prerequisites for the characteristics in the following table.


Characteristic	Prerequisite
Data transformation	If the data in the source file requires transformation, create and compile a transformation formula using the Batch Loader formula type. Use the Manage Fast Formulas task in the Payroll Calculation work area.

Characteristic	Prerequisite
File encryption	<p>Before loading encrypted files, ensure that encryption keys exist for secure file transfer. This involves creating a service request, generating PGP key pairs, and sharing the encryption keys.</p> <p>Specify the Payroll Batch Loader Encryption Type parameter value for the process configuration group you select when running the flow or the web service. Valid values are PGPSIGNED, PGPUNSIGNED, and NONE. Use the Manage Payroll Process Configuration task in the Setup and Maintenance work area to specify processing parameters for your process configuration group.</p>
Other processing parameters	Use the Manage Payroll Process Configuration task in the Setup and Maintenance work area to add parameters for the process configuration group, as needed. Examples of processing parameters are Batch Error Mode, Logging Area, Logging Category, and Threads.

Creating the Batch Manually

The following steps describe how to submit a flow manually to create the batch from a file that you have already uploaded:

1. Use the Submit a Payroll Flow task in the Checklist work area, and select the **Load Batch from File** flow pattern.
2. Enter a name for the flow.
3. If the source file already exists on the content server, enter the content ID, otherwise, leave the Content ID field blank.
4. Enter a name for the batch.
5. Optionally, select the transformation formula and process configuration group.
6. Submit the flow and click **OK and View Checklist**.
7. If you supplied the content ID, select the row for Upload File, and then right-click to select **Mark as Complete**. Otherwise, if you are uploading your file now, perform these steps:
 - a. In the row for Upload File, click **Go to Task**.
 - b. Click **Manage Attachments**.
 - c. Click **Browse** to select your file to upload, and then click **OK**.
 - d. Click **Done**.
8. Click **Refresh** until the Upload File task and the Load Batch task show as complete.
9. Verify the batch content as follows:
 - a. In the row for the Verify Batch task, click **Go to Task**.
 - b. Download and open the batch loader workbook.
 - c. Select your legislative data group, and then click **Search**.
 - d. Double-click the batch name that you entered when submitting the Load Batch From File flow.
 - e. On the Batch Content Sheet, double-click the task action name, for example, **Add Payroll**.
 - f. Verify the content and save the workbook to upload any changes you made.
10. Transfer the batch as follows:
 - a. Use the Submit a Process or Report task in the Checklist or Payroll Administration work area, and select the **Transfer Batch** flow pattern.
 - b. Enter a name for the flow.
 - c. Search for and select the batch to transfer.
 - d. Submit the flow.

 **Tip:** To roll back a batch of element entries, use the Roll Back Batch Transfer process from the Payroll Administration work area. The Roll Back Batch Transfer process supports only batches using the Create Element Entry task.

Automating Batch Creation

You can combine the Create Batch task and the Transfer Batch task into a custom flow pattern. You can then use the Flow Actions web service to call your custom flow for automation. Specify a transformation formula to transform the data, as needed.

Related Topics

- [Setting up Encryption for File Transfer: Procedure](#)
- [Payroll Process Configuration Parameters](#)
- [SOAP Web Services for Oracle HCM Cloud](#)

Automating Payroll Data Loads: Procedure

You can automate the process to load payroll-related data, such as element entries, on a schedule of your choice using the Flow Actions Service web service. You write your own programs to load the source file to Oracle WebCenter Content Server and invoke the Flow Actions web service that submits a custom flow. The custom flow loads the file into a batch using the payroll batch loader, and transfers the batch from the staging tables into the application tables.

To automate data loading, you must:

- Copy the Load Batch from File flow pattern and edit it for automation.
- If you have set up a security profile, add the new flow pattern to the security profile.
- Create the program to upload files to the content server and invoke the Flow Actions Service web service.

Optionally, when submitting the flow, you must specify:

- A process configuration group as a flow parameter, if your source data is encrypted or if you require other processing parameters.
- A file transformation formula, if the source file contains data that must be transformed, such as deriving an assignment number from a person number.

Copying and Editing the Flow Pattern

You copy and edit the predefined flow pattern to:


- Remove the manual tasks for uploading the file and verifying the batch.
- Add the Transfer Batch task so the transfer automatically follows the loading of the file to the staging tables.

Complete the following steps:


1. Select the **Manage Payroll Flow Patterns** task in the Payroll Checklist work area.
2. Search for and select **Load Batch from File**, and then click **Copy**.
3. In the Copy Flow Pattern window, complete the fields as shown in this table:

Field	Value
Name	Enter the same name that you use to identify the flow pattern in the flow action service.
Legislative Data Group	Optionally, select a legislative data group to restrict access to this flow.

- Click **Save and Close**.
- Search for and select the new flow pattern, and then click **Edit**.

 **Note:** If you selected a legislative data group when making the copy, you must select it when searching for the new flow pattern.

- Remove two flow tasks as follows:
 - Select **Upload File**, click **Delete**, and then click **Yes**.
 - Select **Verify Batch**, and then click **Delete**.
- Add the Transfer Batch flow task as follows:
 - Click **Select and Add**.
 - Search for and select **Transfer Batch**, and then click **Done**.
- On the Task Sequence page, review the Task column and the Following Task column and ensure that the tasks are in this sequence: Start Flow, Load Batch, Transfer Batch, End Flow.
- Optionally, on the Parameters page, select and edit the flow parameters as shown in this table.

 **Note:** If you are using a transformation formula with preconfigured process configuration group parameters, you already have values to use by default, so you don't need this step.

Parameter Name	Parameter Basis	Basis Value
Process Configuration Group	Post SQL Bind	<pre>select action_param_group_id from pay_action_param_groups where action_param_group_name = 'Configuration Group Name that has the encryption type to be used'</pre>
Transformation Formula	Post SQL Bind	<pre>select max(formula_id) from ff_formulas_vl ff, ff_formula_types ft where ff.formula_type_id = ft.formula_type_id and ft.formula_type_name = 'Batch Loader' and ff.formula_name = 'Name of formula to be used'</pre>

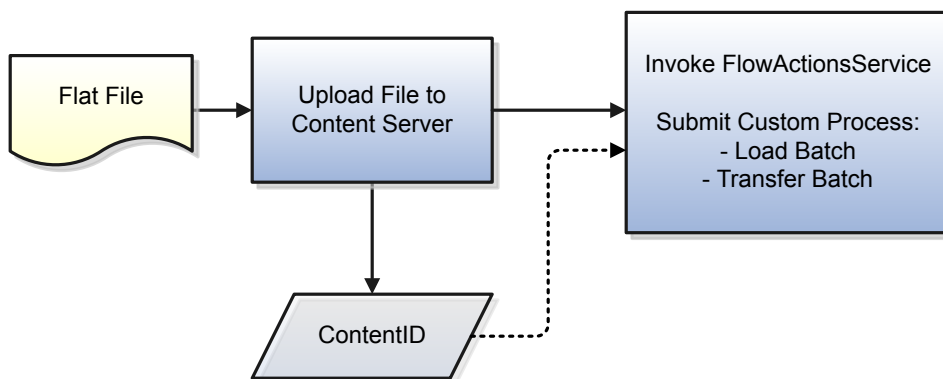
10. Edit the Transfer Batch task as follows:
 - a. On the Tasks tab, select **Transfer Batch**, click **Edit**, and then click **Edit Task**.
 - b. On the Basic Information page, select the **Batch** parameter.
 - c. In the Batch Parameter Details section, in the **Parameter Basis** field, select **Bind to Flow Task**.
 - d. In the **Basis Value** field, select **Batch**, and then click **Next**.
 - e. Optionally, enter ownership information, and then click **Next**.
 - f. Optionally, enter duration and notification information, and then click **Submit**.
11. Edit the Load Batch task as follows:
 - a. On the Tasks tab, select **Load Batch**, click **Edit**, and then click **Edit Task**.
 - b. On the Basic Information page, select the **Batch** parameter.
 - c. In the Batch Parameter Details section, in the **Usage** field, select **Output parameter**, and then click **Next**.
 - d. Optionally, enter ownership information, and then click **Next**.
 - e. Optionally, enter duration and notification information, and then click **Submit**.
12. On the Manage Payroll Flow Patterns page, click **Submit**.

Adding the Flow Pattern to a Security Profile

The HCM data role security controls who can submit the flow pattern or view the resulting flow from the Payroll Dashboard or payroll work areas. If a security profile exists, an IT security manager must add the new flow pattern to the security profile.

Creating the Program for Automation

Your custom program uploads the source file to the content server, and retrieves the content ID for the Flow Actions Service web service.



Your program then calls the Flow Actions web service supplying the following parameters:

- Name of the custom flow pattern (copy of Load Batch From File)
- Content ID of the uploaded file
- Unique name to identify the flow instance being submitted
- Unique name for batch to create and then transfer
- Process configuration group name for special processing (optional)

- Transformation formula name (optional)

For more information about the Flow Actions Service web service, refer to the SOAP Web Services for Oracle HCM Cloud guide. For examples of its usage for automating file uploads, refer to the attachment for HCM Data Loader User Guide (1664133.1) on My Oracle Support at <https://support.oracle.com>.

Related Topics

- [Setting up Encryption for File Transfer: Procedure](#)
- [Payroll Process Configuration Groups: Explained](#)
- [HCM Security Profiles: Explained](#)
- [SOAP Web Services for Oracle HCM Cloud](#)

File Format and Data Transformation for Payroll Batch Uploads: Critical Choices

The Load Batch from File process takes data from a flat file and moves that data into a payroll batch loader workbook for upload. The requirements of the format of your flat file vary depending on whether you will transform the type or sequence of data in your file. This topic describes the different file format requirements.

All columns in the batch loader workbook for the task and task name you want to use must be represented in your flat file, even if they are optional in the workbook.

File Format with No Transformation (Default Format)

If the attributes and sequence in the source file are the same as what's expected in the workbook for the task and task action you want, then you don't need any transformation. The file must include the task name, task action name, and data for each record you want to upload.

The following example shows the content of a file that adds three people to different payrolls in the US LDG by assignment number.

```
Payroll Relationship|Add Payroll||E34567|Monthly Payroll|2015-01-01|US LDG
Payroll Relationship|Add Payroll||E5897|Weekly Payroll|2015-01-01|US LDG
Payroll Relationship|Add Payroll||E9736456|Monthly Payroll|2015-01-01|US LDG
```

Notice that it includes the task name, the task action name, and no reference name before the values to upload for each record. The reference name is blank because it isn't applicable in this example. Other task actions, such as Create Element Entry, use the value for reference name is typically the element name.

Files with Data Transformation

Use a simple formula for basic transformation of data, such as changing the order of attributes in your file to the order that the batch loader workbook requires. Use more complex formulas to convert attributes, such as a person number into an assignment number.

The following example shows the content of a file that adds three people to different payrolls by assignment number. Notice that the data, payroll name, and assignment number are in a different sequence from the batch loader workbook, so you would use a transformation formula to change the positions. Because the formula already specifies the task and task action, the file format can begin directly with a column attribute value.

```
2015-10-10|Monthly Payroll|E34567
2015-10-10|Weekly Payroll|E5897
2015-10-10|Monthly Payroll|E9736456
```

Transformation Formulas

The Batch Loader formula must contain values for at least one task name and task action. You can potentially load different types of data, using different tasks, with the same formula and a single batch.

The following example specifies the task name in the formula as Payroll Relationship and the task action name as Add Payroll.

```
/*Batch Related Outputs*/
TASK='Payroll Relationship'
TASKACTION='Add Payroll'
```


The following example specifies the task name in the formula as Element Entry, the task action name as Create Element Entry, and the task action reference as Pay Period Amount.

```
/*Batch Related Outputs*/
TASK='Element Entry'
TASKACTION='Create Element Entry'
ACTIONREFERENCE='Pay Period Amount'
```

Delimiters

The default delimiter that separates values is a pipe character. If your file uses a different delimiter, you must set the delimiter you want in a Batch Loader transformation formula. The following example specifies a comma character as the delimiter.

```
/* Calculations */
IF OPERATION='FILETYPE' THEN
  OUTPUTVALUE='DELIMITED'
ELSE IF OPERATION='DELIMITER' THEN
  OUTPUTVALUE=', '
```

 **Note:** Ensure that the delimiter you enter in the formula is a single non-ASCII character and not part of any of the values to upload.

Batch Loader Formula Type

The Batch Loader formula type controls how the application retrieves information when loading a batch from a file. You create batch loader formulas on the Manage Fast Formulas page when the file you want to upload requires transformation.

Examples of why you might create a batch loader formula include:

- To assign the position of attributes in the file because they are not in the expected sequence.
- To convert an attribute in the file to another attribute that you derive using value sets.
- To use a single formula for separate batch loader tasks.

Contexts

The following contexts are available to all formulas of the Batch Loader type:

- EFFECTIVE_DATE (Text)
- TASK (text)

Name of the task from the batch loader workbook.

- TASKACTION (Text)

Name of the task action from the batch loader workbook.


- REFERENCE (Text)

The name of the element for task actions such as element entries, costing setups, and balance adjustments.

Database Items

The Batch Loader formula type does not support database items. You can use database items in your batch loader formulas by calling a formula within your formula.

To retrieve information when a database item isn't available, use the GET_VALUE_SET function.

 **Note:** When using the GET_VALUE_SET function, ensure that the Value Attributes Table Alias field for the value set has no value. The function will fail when an alias is provided.

Input Variables

The following input variables are available for all formulas of Batch Loader formula type. Additional variables may be available depending on the selected task and task action.

Input and Data Type	Required	Notes
OPERATION (text)	Yes	Available operations: <ul style="list-style-type: none">• FILETYPE If specified, value must be DELIMITED.• DELIMITER The character used to delimit the columns in the file.• MAP Where the information for the task and task action is mapped from the file content.
PROCESS_DATE (text)	No	Must be in the format YYYY-MM-DD HH:MM:SS where hours represent a 24-hour clock.
LEGISLATIVE_ DATA_GROUP_ID (Text)	No	
LINESEQUENCE (Number)	No	Line number to be used for the batch. The following example takes the line number and multiplies by 10; the created batch lines will increment from 10, 20, 30, and so on.
LINEREPEATNO (Number)	No	Number of times the formula executes for the given line in the file.

Input and Data Type	Required	Notes
LINEREPEAT (Number)	No	Enable creating multiple batch lines for a single row in a file. Valid values are Y and N. If set as Y, after creating the batch line the formula is called again for the same file content. LINEREPEATNO indicates how many repetitions. For example, for time entry, there might be a regular time entry wage followed by a premium time entry wage.
TOTALLINES (Number)	No	Total number of lines in the file. Use, for example, to compare a hard-coded total in the file to reconcile the file content.
POSITION1 (Text)	Yes	First position of the column in the file.
POSITION2 (Text)	Yes	Second position of the column in the file.
POSITION3-POSITION30 (Text)	No	Optional additional positions of columns in the file.
TOTALLINES (Number)	No	For example, to compare a hard-coded total in the file to reconcile the file content.

Return Values

The return values for batch loader formulas vary based on the business object and task action. Return values are the same as the parameter names with space characters replaced by underscore characters.


Sample Formula for Batch Loader File Transformation

The transformation formula in this example specifies the position of attributes because the attributes in the flat file are in a different sequence than the sequence expected in payroll batch loader workbook. The formula also converts Person Number in the flat file into Assignment Number.

The formula's return values are the same as the parameter name with space characters replaced by underscore characters.

The following snippet in the formula assigns the attributes to the required position and performs the conversion using values sets to derive the necessary values:

```
/*Task Action Related Outputs */  
PAYROLL=POSITION2  
ASSIGNMENT_NUMBER=GET_VALUE_SET('SAMPLE_GET_ASG_NUM','|=PERSON_NUMBER=''||POSITION3||''')  
Effective_As_Of_Date=POSITION1
```

 **Tip:** To debug value sets, create a BI report with the following query as a data model to return the required data: `SELECT pay_ff_functions.gvs ('SAMPLE_GET_ASG_NUM','|=PERSON_ID=100000012092216') value FROM dual;`

The complete formula for this example is as follows:

```

/*****
FORMULA NAME: Transform Attribute - Payroll Relationship
FORMULA TYPE: Batch Loader
Sample Input File Format:
Effective As Of Date|Payroll|Person Number/

/* Inputs */
INPUTS ARE OPERATION (text), LINESEQUENCE (number), LINEREPEATNO (number),
POSITION1 (text), POSITION2 (text), POSITION3 (text)

DEFAULT FOR LINENO IS 1
DEFAULT FOR LINEREPEATNO IS 1
DEFAULT FOR LINESEQUENCE IS 1
DEFAULT FOR POSITION1 IS 'NO DATA'
DEFAULT FOR POSITION2 IS 'NO DATA'
DEFAULT FOR POSITION3 IS 'NO DATA'

/* Calculations */
IF OPERATION='FILETYPE' THEN
OUTPUTVALUE='DELIMITED'
ELSE IF OPERATION='DELIMITER' THEN
OUTPUTVALUE='|'
ELSE IF OPERATION='MAP' THEN
(
/*Batch Related Outputs*/
TASK='Payroll Relationship'
TASKACTION='Add Payroll'

/*Task Action Related Outputs */
PAYROLL=POSITION2 ASSIGNMENT_NUMBER=GET_VALUE_SET('SAMPLE_GET_ASG_NUM','|=PERSON_NUMBER='''||POSITION3|''')
Effective_As_Of_Date=POSITION1
)
ELSE
OUTPUTVALUE='NONE'

/* Returns */
IF OPERATION='MAP' THEN
RETURN TASK,TASKACTION,LINESEQUENCE,PAYROLL,ASSIGNMENT_NUMBER,Effective_As_Of_Date
ELSE
RETURN OUTPUTVALUE
/* End Formula Text */

```

Sample Batch Loader Formula for Multiple Tasks

The formula in this example uses two separate tasks to create bank and also add payroll to a payroll relationship. The formula also converts Person Number in the flat file into Assignment Number.

The formula for this example is as follows:

```

/*****
FORMULA NAME: Create Bank and Payroll Relationship
FORMULA TYPE: Batch Loader
Sample Input File Format:
Effective As Of Date|Payroll|Assignment Number
*****/

/* Inputs */
INPUTS ARE OPERATION (text),
LINEREPEATNO (number), POSITION1 (text), POSITION2 (text), POSITION3 (text), POSITION4 (text),

```

```
POSITION5 (text),

DEFAULT FOR LINESEQUENCE IS 1
DEFAULT FOR LINEREPEATNO IS 1
DEFAULT FOR POSITION1 IS 'NO DATA'
DEFAULT FOR POSITION2 IS 'NO DATA'
DEFAULT FOR POSITION3 IS 'NO DATA'
DEFAULT FOR POSITION4 IS 'NO DATA'
DEFAULT FOR POSITION5 IS 'NO DATA'

/* Calculations */
IF OPERATION='FILETYPE' THEN OUTPUTVALUE='DELIMITED'
ELSE IF OPERATION='DELIMITER' THEN OUTPUTVALUE='|'
ELSE IF OPERATION='READ' THEN OUTPUTVALUE='NONE'
ELSE IF OPERATION='MAP' THEN
(

/*Batch Related Outputs*/
IF POSITION1 = 'Payroll Relationship' AND POSITION2 = 'Add Payroll' THEN (
TASK='Payroll Relationship' TASKACTION='Add Payroll'

/*Task Action Related Outputs */
PAYROLL=POSITION4
ASSIGNMENT_NUMBER=POSITION5
Effective_As_Of_Date=POSITION3
)
IF POSITION1 = 'Bank and Branch' AND POSITION2 = 'Create Bank' THEN(
TASK='Bank and Branch'
TASKACTION='Create Bank'

/*Task Action Related Outputs */
Bank_Name=POSITION3
)
)
ELSE OUTPUTVALUE='NONE'

/* Returns */
IF OPERATION='MAP'
THEN IF POSITION1 = 'Payroll Relationship' AND POSITION2 = 'Add Payroll' THEN (
RETURN TASK,TASKACTION,LINSEQUENCE,PAYROLL,ASSIGNMENT_NUMBER,Effective_As_Of_Date
)
IF POSITION1 = 'Bank and Branch' AND POSITION2 = 'Create Bank' THEN (
RETURN TASK,TASKACTION,LINSEQUENCE,Bank_Name
)
ELSE RETURN OUTPUTVALUE


/* End Formula Text */
```


23 Payroll Batch Loader Workbooks

Balance Adjustments Workbook

Import data for balance adjustments using the Batch Loader task in the Payroll Administration, Data Exchange, or Checklist work area. You enter the data into integrated Microsoft Excel workbooks, save it to the staging tables, and then transfer the data into the application. Using the payroll batch loader, you can upload multiple balance adjustments at the same time in a single batch. This topic explains the columns in the workbook for the Adjust Balance task action.

When you add the Adjust Balance task action, you enter **Balance** into the Task Name field, and select the element whose balances you want to adjust in the Reference field. You can upload multiple balance adjustments for only one element per batch. After saving the batch content, you submit the Transfer Batch process to move the data to the application tables, and then submit the Adjust Multiple Balances process to perform the adjustment. Columns will vary depending on the legislative data group and the selected element.

 **Note:** The batch loader doesn't support balance adjustments to different payrolls or different inclusion rules within the same batch. Each of the following columns must have the same value: Effective Date, Payroll, Include in Costing Process, Include in Payment Balance: If you have different values for these columns, you must create a separate batch for each combination.

Adjust Balance

The workbook for the Adjust Balance task action uses the following columns to upload data for balance adjustments.


Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective Date	Yes	DATE The date that is used for the balanced adjustment. Must be in the format YYYY-MM-DD. Enter the same effective date for all rows in the batch.
Payroll	Yes	VARCHAR2(80) Name of the payroll definition of the balance adjustment. Enter the same payroll definition for all rows in the batch. Payroll definition names are case-sensitive. Enter the name exactly.

Column	Required	Comments
Consolidation Group	Yes	<p>VARCHAR2(60)</p> <p>Name of the existing consolidation group for the specified payroll definition. Enter the same consolidation group for all rows in the batch.</p> <p>Consolidation group names are case-sensitive. Enter the name exactly.</p>
Balance Date	No	<p>DATE</p> <p>The balance date, if the balance dimension requires an entry. Otherwise, leave blank.</p> <p>If specified, must be in the format YYYY-MM-DD.</p>
Assignment Number	Yes	<p>NUMBER(18)</p> <p>Existing assignment number of the person whose balance you want to adjust.</p>
Calculation Breakdown Component	Yes	<p>VARCHAR2(80)</p> <p>Name of the calculation breakdown component as on the Adjust Individual Balances page. This is typically the tax reporting unit associated to the calculation card for the specified person.</p>
Tax Reporting Unit	Yes	<p>VARCHAR2(80)</p> <p>Name of the tax reporting unit for the payroll. It is best practice to use a single tax reporting unit for all rows in the batch.</p>
Legal Employer	No	<p>VARCHAR2(80)</p> <p>Name of the legal employer for the payroll. It is best practice to use a single legal employer for all rows in the batch.</p>
Third-Party Payee	No	<p>VARCHAR2(80)</p> <p>Name of the third-party payee, if the adjustment for a payment balance used to pay a third-party organization, such as a court order.</p>
Time Definition	No	<p>VARCHAR2(80)</p> <p>The name of the time definition for the balance to adjust.</p>
Include in Costing Process	No	<p>VARCHAR2(30)</p>

Column	Required	Comments
		Whether to include the balance adjustment in costing processes. Valid values are Yes and No.
		Enter the same value for all rows in the batch. If not provided, the value used by default is No.
Input Value 1 - Input Value 30	No	VARCHAR2(250) Varying input value column names based on element-specific attributes and the element's legislative data group.
Segment 1 - Segment 30	No	VARCHAR2(250) Costing segments based on the Cost Allocation key flexfield.
Include in Payment Balance	No	VARCHAR2(30) Whether to include the balance adjustment in payment balances. Valid values are Yes and No. Enter the same value for all rows in the batch. If not provided, the value used by default is No.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

```
Balance|Adjust Balance|Element Name|Effective Date|Payroll|Consolidation Group|Balance Date|Assignment
Number|Calculation Breakdown Component|Tax Reporting Unit|Legal Employer|Third-Party Payee|Time Definition|
Include adjustment in costing process|Input Value 1|Input Value 2|Input Value 3|Input Value 4|Input Value 5|
Input Value 6|Input Value 7|Input Value 8|Input Value 9|Input Value 10|Input Value 11|Input Value 12|Input
Value 13|Input Value 14|Input Value 15|Input Value 16|Input Value 17|Input Value 18|Input Value 19|Input
Value 20|Input Value 21|Input Value 22|Input Value 23|Input Value 24|Input Value 25|Input Value 26|Input
Value 27|Input Value 28|Input Value 29|Input Value 30|Segment 1|Segment 2|Segment 3|Segment 4|Segment 5|
Segment 6|Segment 7|Segment 8|Segment 9|Segment 10|Segment 11|Segment 12|Segment 13|Segment 14|Segment 15|
Segment 16|Segment 17|Segment 18|Segment 19|Segment 20|Segment 21|Segment 22|Segment 23|Segment 24|Segment
25|Segment 26|Segment 27|Segment 28|Segment 29|Segment 30|Include in payment balance
```

 **Note:** Input Value 1 through 30 are based on the element. Similarly, Segment 1 through 30 are based on the cost structure.

Related Topics

- [Adjusting Balances: Critical Choices](#)
- [Adjusting Multiple Balances: Procedure](#)
- [Importing Data Using the Payroll Batch Loader: Explained](#)

Balance Groups Workbook

Import balance groups using the Batch Loader task in the Payroll Administration, Data Exchange, or Checklist work area. You enter the data in an integrated Microsoft Excel workbook, save it to the staging tables, and then transfer the data into the application. This topic explains the task actions you can add to the workbook for the Balance Group task and the columns you complete for each task action.

Using the payroll batch loader, you can create, update, or delete multiple balance groups for multiple legislative data groups at the same time. This table describes the batch upload tasks that manage payroll definitions.

Task Action	Purpose
Add a Defined Balance	Add a balance definition to a balance group, which can be used to retrieve balance values for reporting, page displays, and payroll processes.
Remove a Balance	Remove an unused balance group definition.

Add a Defined Balance

The Add a Defined Balance task workbook uses the following columns to add balance definitions.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Attribute Definition	Yes	VARCHAR2(240) Attribute definitions identify separate components of a component group that are used in the application or referenced in formulas.
Balance Dimension	Yes	VARCHAR2(160) Balance dimensions describe the scope of a balance value, such as the period of time over which it is accumulated and whether it relates to a single assignment, terms, or payroll relationship.
Balance Type	Yes	VARCHAR2(120) Name of the balance. Balance names are case-sensitive. Ensure that you enter the name exactly.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Balance Group|Add a Defined Balance||*Attribute Definition|*Balance Dimension|*Balance Type

Remove a Defined Balance

The Remove a Defined Balance task workbook uses the following columns to delete an unused balance definition.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Attribute Definition	Yes	VARCHAR2(240) Attribute definitions identify separate components of a component group that are used in the application or referenced in formulas.
Balance Dimension	Yes	VARCHAR2(160) Balance dimensions describe the scope of a balance value, such as the period of time over which it is accumulated and whether it relates to a single assignment, terms, or payroll relationship.
Balance Type	Yes	VARCHAR2(120) Name of the balance. Balance names are case-sensitive. Ensure that you enter the name exactly.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Balance Group|Remove a Defined Balance||*Attribute Definition|*Balance Dimension|*Balance Type

Related Topics

- [Importing Data Using the Payroll Batch Loader: Explained](#)

Bank Data Workbooks

Import bank data using the Batch Loader task in the Payroll Administration, Data Exchange, or Checklist work area. You enter the data in integrated Microsoft Excel workbooks, save it to the staging tables, then transfer the data into the application. This topic explains the task actions you can add to the workbook for the Bank and Branch task and External Bank Account task.

This table explains the task actions that you can add to the workbook that load bank information.

Task Name	Task Action	Purpose
Bank and Branch	Create Bank	Create a bank name and optional bank code, making it available when creating bank branches.
Bank and Branch	Inactivate Bank	Inactivate a bank in a specified country, including its branches and internal and external accounts.
Bank and Branch	Create Bank Branch	Create a branch of an existing bank. Branch data includes name, number, and identifiers for electronic funds transfer (EFT) payments.
Bank and Branch	Inactivate Bank Branch	Inactivate a bank branch, including any internal and external accounts.
External Bank Account	Create External Bank Account	Create a bank account, based on an existing bank and branch, to use in personal payment methods. You can't create internal bank account numbers of source accounts for payments to workers using this task.
External Bank Account	Inactivate External Bank Account	Inactivate an external bank account used to receive EFT payments for workers.
External Bank Account	Add Person to Existing Account	Associate workers or third-party people to existing bank accounts to enable more than one person to receive EFT payments into the same bank account.

These tasks have interdependencies. Create separate workbooks for each task, for each legislative data group where you add bank information, to preserve the data dependency. For example, you can first create multiple banks in one workbook, and then create all of the branches in the next workbook, and so on.

Create Bank

The Create Bank task action uses the following columns to create a bank name and optional bank code identifier.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Country Code	Yes	VARCHAR2(60) Existing code for the country or territory of the bank to create.

Column	Required	Comments
Bank Name	Yes	<p>VARCHAR2(1440)</p> <p>Name of the bank to create.</p> <p>Ensure that no bank with the same name exists in your database and that you follow all naming conventions.</p>
Bank Number	No	<p>VARCHAR2(400)</p> <p>Bank number of bank to create. Bank number validation varies depending on country-specific rules.</p>

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Bank and Branch|Create Bank|*Country Code|*Bank Name|Bank Number

Inactivate Bank

The Inactivate Bank task action uses the following columns to inactivate an existing bank with the specified country code.

Column	Required	Comments
Line Sequence	Yes	<p>NUMBER</p> <p>Enter 1 for the first row and continue sequentially for subsequent rows.</p>
Country Code	Yes	<p>VARCHAR2(60)</p> <p>Existing code for the country or territory of the bank to inactivate.</p>
Bank Name	Yes	<p>VARCHAR2(1440)</p> <p>Name of the existing bank to inactivate. Ensure that the name is an exact match with the name that exists in your database.</p>

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Bank and Branch|Inactivate Bank||*Country Code|*Bank Name

Create Bank Branch

The Create Bank Branch task action uses the following columns to create branch information for a specified bank name.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Country Code	Yes	VARCHAR2(60) Existing code for the country or territory of the branch to create.
Bank	Yes	VARCHAR2(1440) Name of the existing bank for the branch to create. Bank names are case-sensitive. Enter the name exactly.
Branch Name	Yes	VARCHAR2(1440) Name of branch to create. Must be unique for the bank name and legislative data group that you select in the batch header.
Branch Number	Yes	VARCHAR2(120) Branch number of branch to create. Must be unique for the bank name and legislative data group that you select in the batch header. Branch number validation varies depending on country-specific rules. For example, in Australia, the combined value of bank number and branch number must not exceed six numbers.
BIC/SWIFT Code	No	VARCHAR2(120) Bank identifier code or SWIFT code that identifies bank and branch information for payments between two financial institutions.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Bank and Branch|Create Bank Branch||*Country Code|*Bank|*Branch Name|Branch Number|BIC/Swift Code

Inactivate Bank Branch

The Inactivate Bank Branch task action uses the following columns to inactivate an existing bank and branch with the specified country code.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Country Code	Yes	VARCHAR2(60) Existing code for the country or territory of the bank to inactivate.
Bank Name	Yes	VARCHAR2(1440) Name of the existing bank whose branch to inactivate. Ensure that the name is an exact match with the name that exists in your database.
Branch Name	Yes	VARCHAR2(1440) Name of the branch to inactivate. Must be unique for the bank name and legislative data group that you select in the batch header. Ensure that the name is an exact match with the name that exists in your database.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Bank and Branch|Inactivate Bank Branch||*Country Code|*Bank Name|*Branch Name

Create External Bank Account

The Create External Bank Account task action uses the following columns to create bank accounts, based on existing banks and branches. After you create external bank accounts, you can use them in personal payment methods for workers and third-party payees.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Bank Name	Yes	VARCHAR2(1440) Name of existing bank. Bank names are case-sensitive. Enter the name exactly.
Bank Branch Name	Yes	VARCHAR2(1440) Name of existing branch. Bank branch names are case-sensitive. Enter the name exactly.

Column	Required	Comments
Account Number	Yes	NUMBER(18) The bank account number to receive EFT payments.
IBAN	No	VARCHAR2(200) International bank account number conforming to the ISO standard for uniquely identifying a bank account for payments between banks. Applies only for certain countries or territories.
Account Type	No	CHAR(32) Based on values in the AR_IREC_BANK_ACCOUNT_TYPES lookup table. Valid values are: <ul style="list-style-type: none"> CHECKING MONEYMRKT SAVINGS UNKNOWN
Secondary Account Reference	No	VARCHAR2(120) Usage varies by country or territory, for example, Building Society Number in the UK.
Account Name	No	VARCHAR2(1440) Label used to identify bank account when there are multiple accounts, for example, Checking or Savings.
Person Number	Yes	NUMBER(18) Payroll relationship ID or third-party payee ID of an existing person with a corresponding TCA party.
Country	No	VARCHAR2(60) Existing code for the country or territory of the bank whose external account to create.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

External Bank Account|Create External Bank Account||*Bank Name|*Bank Branch Name|*Account Number|IBAN|Account Type|Secondary Account Reference|Account Name|*Person Number|Country

Inactivate External Bank Account

The Inactivate External Bank Account task action uses the following columns to inactivate an existing external bank account with the specified country, bank, and branch.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Country Code	Yes	VARCHAR2(60) Existing code for the country or territory of the bank whose external account to inactivate.
Bank Name	Yes	VARCHAR2(1440) Name of an existing bank whose external account to inactivate. Bank names are case-sensitive. Enter the name exactly.
Bank Branch Name	Yes	VARCHAR2(1440) Name of the existing branch whose bank account to inactivate. Branch names are case-sensitive. Enter the name exactly.
Account Number	Yes	NUMBER(18) Bank account number to inactivate.


If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

External Bank Account|Inactivate External Bank Account||*Country Code|*Bank Name|*Bank Branch Name|*Account Number

Add Person to Existing Account

The Add Person to Existing Account task action uses the following columns to enable multiple people to receive EFT payments into the same bank account.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.

Column	Required	Comments
Country Code	Yes	VARCHAR2(60) Existing code for the country or territory of the external account.
Bank Name	Yes	VARCHAR2(1440) Name of existing bank. Bank names are case-sensitive. Enter the name exactly.
Account Number	Yes	NUMBER(18) The existing bank account number to receive EFT payments.  Tip: Create bank account numbers using the Create External Bank Accounts task action.
Person Number	Yes	NUMBER(18) Payroll relationship ID or third-party payee ID of an existing person.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

External Bank Account|Add Person to Existing Account||*Country Code|*Bank Name|*Account Number|*Person Number


Costing Setups Workbook

Import costing setups using the HCM Data Loader. Alternatively, for small batches of data, you can use the batch loader to import costing setup information from integrated workbook templates into staging tables and transfer that data into the application. This topic explains the task actions for the Cost Allocation task and the columns you complete for each task action.

Create or delete costing details for the cost and offset accounts by using the Create Cost Allocation and Delete Cost Allocation tasks. You can import costing setups for cost and offset accounts, but not costing setups for the suspense, default, and priority accounts. For costing setups for element entries, use the Element Entry batch loader task, and the Create Element Entry task action. You cannot delete costing setups for element entries.

Cost Allocation Columns

The cost allocation information determines the settings to apply and the account numbers to enter at each level of the cost hierarchy for your cost allocation key flexfield. Payroll uses this information to derive the costing results. If you aren't submitting the Calculate Payroll process, you can extract the costing setup information for your third-party payroll provider. The Delete Cost Allocation task uses the same columns as the Create Cost Allocation task.

 **Note:** Loading costing entries using the payroll batch loader is intended for initial implementation. Use caution when loading additional costing setups to avoid creating duplicate records. The batch loader performs no validation against existing costing setup objects.

The batch loader is case-sensitive. Ensure that you enter values exactly. Enter object and element names exactly as they appear in the application. To identify the source object names, you can query the costing objects on the costing setup pages in the Accounting Distribution work area.

Column	Required	Comments
Line Sequence	Yes	<p>NUMBER</p> <p>Enter 1 for the first row and continue sequentially for subsequent rows.</p>
Effective Start Date	Yes	<p>DATE</p> <p>Enter the first date that processes can use the costing information. Use a date on or before the earliest date of any historical data you load.</p> <p>Must be in the format YYYY-MM-DD for all of the dates.</p>
Effective End Date	Yes	<p>DATE</p> <p>Enter the last date that processes can use the costing information. Use a date later than the Effective Start Date.</p>
Source Type	Yes	<p>TEXT(60)</p> <p>Enter the kind of source object to cost.</p> <p>Each source type requires specific information in the Source and Creator columns. Refer to the table later in this topic that lists the valid values for objects based on the cost hierarchy.</p> <p>To cost a person's element at a specified level, use one of the following valid values:</p> <ul style="list-style-type: none"> • AET (assignment) • PRET (payroll relationship) • TET (employment terms) <p>To cost elements of a person at a specified level, use one of the following valid values:</p> <ul style="list-style-type: none"> • ASG (assignment) • PREL (payroll relationship) • TERM (employment terms)


Column	Required	Comments
		<p>To cost a specific object, use one of the following valid values:</p> <ul style="list-style-type: none"> • POS (position) • JOB (job) • ORG (organization) • EL (element eligibility, which uses as the input value the special purpose output value) • LIV (input value for the element eligibility other than the special purpose output value) • PAY (payroll)
Source	Yes	<p>TEXT(60) or NUMBER(18)</p> <p>Enter a value that is compatible with the value in the Source Type column.</p> <p>For example, enter the name of the specific source object to cost, or to cost all the elements for a person, the person's assignment number, payroll relationship number, or the employment terms number.</p>
Creator	No	<p>TEXT(60) or NUMBER(18)</p> <p>Required only to cost a job, an input value, or a person's element at a specified level. Use one of the following valid values:</p> <ul style="list-style-type: none"> • Job name or job set name • Input value name • Assignment number, payroll relationship number, or employment terms number
Costing Type	No	<p>TEXT(60)</p> <p>Specify a value only if you are costing an element eligibility record.</p> <p>The kind of costing, such as fixed costing. Valid values are case-sensitive.</p> <ul style="list-style-type: none"> • C (Costed) • F (Fixed Costed) • D (Distributed) • N (Not Costed) <p>Use the Manage Common Lookups task to see the lookup codes for the PAY_COSTABLE_TYPE lookup type. If you don't create a costing record for an element</p>

Column	Required	Comments
		eligibility record, the result is the same as if you created a record for Not Costed (N).
Distribution Set	No	<p>TEXT(60)</p> <p>If you are entering the costing setup information for an element eligibility record that has a distributed costing type, enter the name of the element distribution object element group.</p>
Transfer to General Ledger	No	<p>TEXT (60)</p> <p>If you are entering costing setup information for an element eligibility record, indicate whether to transfer the element's costing results to general ledger.</p> <p>Valid values are Y (for Yes) and N (for No).</p> <p>Use the Manage Common Lookups task to see the lookup codes for the YES_NO lookup type.</p>
Cost Type	No	<p>TEXT(60)</p> <p>Enter one of the following valid values for an element eligibility record:</p> <ul style="list-style-type: none"> • COST (for a cost account) • BAL (for an account that offsets the costing entry, so that the sum of journal entries balance debits and credits)
Proportion	No	<p>NUMBER</p> <p>Enter a decimal value for a number, which represents the percentage of the costing result to allocate to the specified combination of costing segments.</p> <p>Valid values are greater than 0 and less than or equal to 1, where 1 is 100 percent, such as 0.25 for 25 percent.</p>
Segment 1-30	No	<p>NUMBER</p> <p>Enter an account number for each segment that should be populated for the source object, based on your Cost Allocation key flexfield.</p>

If you are uploading a file for these tasks instead of entering data directly into the workbook, you must use the following format:

Cost Allocation|Create Cost Allocation or Delete Cost Allocation||*Effective Start Date|*Effective End Date|*Source Type|*Source|Creator|Cost Type|Proportion|Segment 1|Segment 2|Segment 3|Segment 4|Segment 5|Segment

6|Segment 7|Segment 8|Segment 9|Segment 10|Segment 11|Segment 12|Segment 13|Segment 14|Segment 15|Segment 16|Segment 17|Segment 18|Segment 19|Segment 20|Segment 21|Segment 22|Segment 23|Segment 24|Segment 25|Segment 26|Segment 27|Segment 28|Segment 29|Segment 30|Costing Type|Distribution Group|Transfer to General Ledger

 **Note:** Segment 1 through 30 are based on the cost structure.

Valid Values for the Source, Source Type, and Creator Columns

When the Source Type requires specific information in the Source and Creator columns, you can use this table to quickly locate the costing setup object and the required details to specify for those columns.

Costing Setup Objects	Source Type Column	Source Column	Creator Column
Person's element at the assignment level	TEXT (60)	TEXT (60)	NUMBER(18)
	Valid value is: AET	Enter the element name.	Enter the assignment number. If the number is preceded by text, enter the text exactly, such as E (for employment or employee assignment).
Person's element at the terms level	TEXT (60)	TEXT (60)	NUMBER(18)
	Valid value is: TET	Enter the element name.	Enter the terms number. If the number is preceded by text, enter the text exactly, such as ET (for employment terms).
Person's element at the payroll relationship level	TEXT (60)	TEXT (60)	NUMBER(18)
	Valid value is: PRET	Enter the element name.	Enter the payroll relationship number.
All elements for a person at the assignment level	TEXT (60)	NUMBER (18)	
	Valid value is: ASG	Enter the assignment number.	
All elements for a person at the terms level	TEXT (60)	NUMBER (18)	
	Valid value is: TERM	Enter the terms number.	
All elements for a person at the payroll relationship level	TEXT (60)	NUMBER (18)	
	Valid value is: PREL	Enter the payroll relationship number.	
Position	TEXT (60)	TEXT (60)	TEXT(60)
	Valid value is: POS	Enter the position name.	Enter the position code when you use the same position name for more than one position.
Job	TEXT (60)	TEXT (60)	TEXT(60)

Costing Setup Objects	Source Type Column	Source Column	Creator Column
	Valid value is: JOB	Enter the job name.	Enter a set name when you use the same job name for more than one job.
Department	TEXT (60) Valid value is: ORG	TEXT (60) Enter the department name.	
Element eligibility	TEXT (60) Valid value is: EL	TEXT (60) Enter the name of the element eligibility record. Costing is created for the special purpose output input value.	
Additional input value to cost, other than the special purpose output value	TEXT (60) Valid value is: LIV	TEXT (60) Enter the name of the element eligibility record.	TEXT(60) Valid value is: Name of the input value. For example, in addition to the primary output value, you might require an entry for hours, days, or other units used for statistical entries. No offsets are created for statistical entries.
Payroll	TEXT (60) Valid value is: PAY	TEXT (60) Enter the payroll name.	

Related Topics

- [Payroll Setup Tasks for Costing Accounts: Critical Choices](#)
- [Cost Hierarchy: Explained](#)
- [Importing Data Using the Payroll Batch Loader: Explained](#)


Element Entries Workbook

Import data for element entries using the HCM Data Loader. Alternatively, for small batches of data, you can use the Batch Loader task in the Payroll Administration, Data Exchange, or Checklist work area. You enter the data in integrated Microsoft Excel workbooks, save it to the staging tables, and then transfer the data into the application. This topic explains the columns in the workbook for the Element Entry task and the columns you complete for each task action.

When you add the create or update task action in the workbook, you select the element in the Reference field whose entry values you want to create or update. You can create and update multiple element entry records for only one element per batch. You can delete element entries for multiple elements per batch.

This table explains the three task actions that you can add to the workbook for element entries.

Task Action	Purpose
Create Element Entry	Upload entry values for a single element at a person's assignment, terms, or payroll relationship level.
Update Element Entry	Update entry values for a single element at a person's assignment, terms, or payroll relationship level.
Delete Element Entry	Delete element entries for one or more elements at a person's assignment, terms, or payroll relationship level.

 **Tip:** To roll back a batch of element entries, use the Roll Back Batch Transfer process from the Payroll Administration, Data Exchange, or Checklist work area.

Create Element Entry

The Create Element Entry task action uses the following columns to upload new element entry values for the element specified in the Reference field. You can create multiple element entries, but for only one element per batch. Some columns may vary depending on the legislative data group and the referenced element.


Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective Start Date	Yes	DATE Must be in the format YYYY-MM-DD.
Effective End Date	No	DATE Must be in the format YYYY-MM-DD.
Person Number	No A value for either this field, Assignment Number, Terms Number, or Relationship Number is required. Don't specify a value here if you have a value for any of the other levels.	NUMBER(18) Existing person number of the person whose element entry you want to create. A value for either this field, Assignment Number, Terms Number, or Relationship Number is required. Don't specify a value here if you have a value for either of the other two levels.
Assignment Number	No	NUMBER(18)

Column	Required	Comments
	A value for either this field, Terms Number, or Relationship Number is required. Don't specify a value here if you have a value for either of the other two levels.	Existing assignment number to create the element entry at the assignment level. Use this column when a person has multiple assignments for the same payroll relationship.
Terms Number	No A value for either this field, Assignment Number, or Relationship Number is required. Don't specify a value here if you have a value for either of the other two levels.	NUMBER(18) Existing employment terms number to create the element entry at the employment terms level. Use this column when a person has multiple employment terms for the same payroll relationship.
Relationship Number	No A value for either this field, Assignment Number, or Terms Number is required. Don't specify a value here if you have a value for either of the other two levels.	NUMBER(18) Existing payroll relationship number to create the element entry at the payroll relationship level.
Payroll	No	VARCHAR2(30) Name of the payroll definition for the element entry to create. Must be a valid payroll for the legislative data group used to create the batch. Payroll definition names are case-sensitive. Enter the name exactly.
Reason	No	VARCHAR2(250) The reason is entered as free text.
Subpriority	No	NUMBER(9) Additional priority used to determine processing order when more than one element has the same priority.
Entry Value 1 - Entry Value 50	No	VARCHAR2(150) Entry value columns vary depending on the referenced element.
Segment 1 - Segment 30	No	VARCHAR2(150) Name of the costing segment for the element entry to create.
Override Entry	No	VARCHAR2(30)

Column	Required	Comments
		Indicates whether the element entry preempts other entries for the same element. Valid values are Yes and No.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

```
Element Entry|Create Element Entry|Element Name|Effective Start Date|Effective End Date|Person Number|
Assignment Number|Terms Number|Relationship Number|Payroll|Reason|Subpriority|Entry Value 1|Entry Value
2|Entry Value 3|Entry Value 4|Entry Value 5|Entry Value 6|Entry Value 7|Entry Value 8|Entry Value 9|Entry
Value 10|Entry Value 11|Entry Value 12|Entry Value 13|Entry Value 14|Entry Value 15|Entry Value 16|Entry
Value 17|Entry Value 18|Entry Value 19|Entry Value 20|Entry Value 21|Entry Value 22|Entry Value 23|Entry
Value 24|Entry Value 25|Entry Value 26|Entry Value 27|Entry Value 28|Entry Value 29|Entry Value 30|Entry
Value 31|Entry Value 32|Entry Value 33|Entry Value 34|Entry Value 35|Entry Value 36|Entry Value 37|Entry
Value 38|Entry Value 39|Entry Value 40|Entry Value 41|Entry Value 42|Entry Value 43|Entry Value 44|Entry
Value 45|Entry Value 46|Entry Value 47|Entry Value 48|Entry Value 49|Entry Value 50|Segment 1|Segment 2|
Segment 3|Segment 4|Segment 5|Segment 6|Segment 7|Segment 8|Segment 9|Segment 10|Segment 11|Segment 12|
Segment 13|Segment 14|Segment 15|Segment 16|Segment 17|Segment 18|Segment 19|Segment 20|Segment 21|Segment
22|Segment 23|Segment 24|Segment 25|Segment 26|Segment 27|Segment 28|Segment 29|Segment 30|Override Entry
```

 **Note:** Entry Value 1 through 50 are based on the element. Similarly, Segment 1 through 30 are based on the cost structure.

Update Element Entry

The Update Element Entry task action uses the following columns to upload element entry values for the element specified in the Reference field. You can update multiple element entries, but for only one element per batch. Some columns may vary depending on the legislative data group and the referenced element.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective As-of Date	Yes	DATE Date as of when to make this update effective. Must be in the format YYYY-MM-DD.
Entry Sequence	No	NUMBER The number representing the position of the element entry to update as shown on the element entry summary page. When not specified, the default value is 1. For elements with multiple entry values, you must provide an entry value.
Person Number	No	NUMBER(18)

Column	Required	Comments
	A value for either this field, Assignment Number, Terms Number, or Relationship Number is required. Don't specify a value here if you have a value for any of the other levels.	Existing person number of the person whose element entry you want to create.
Assignment Number	No	NUMBER(18) Existing assignment number to create the element entry at the assignment level. Use this column when a person has multiple assignments for the same payroll relationship.
Terms Number	No	NUMBER(18) Existing employment terms number to create the element entry at the employment terms level. Use this column when a person has multiple employment terms for the same payroll relationship.
Relationship Number	No	NUMBER(18) Existing payroll relationship number to create the element entry at the payroll relationship level.
Payroll	No	VARCHAR2(30) Name of the payroll definition for the element entry to update. Must be a valid payroll for the legislative data group used to create the batch. Payroll definition names are case-sensitive. Enter the name exactly.
Reason	No	VARCHAR2(250) The reason is entered as free text.
Subpriority	No	NUMBER(9) Additional priority used to determine processing order when more than one element has the same priority.
Segment 1 - Segment 30	No	VARCHAR2(150) Name of the costing segment for the element entry to update.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

```
Element Entry|Update Element Entry|Element Name|Effective As-of Date|Entry Sequence|Person Number|Assignment
Number|Terms Number|Relationship Number|Payroll|Reason|Subpriority|Entry Value 1|Entry Value 2|Entry Value
3|Entry Value 4|Entry Value 5|Entry Value 6|Entry Value 7|Entry Value 8|Entry Value 9|Entry Value 10|Entry
Value 11|Entry Value 12|Entry Value 13|Entry Value 14|Entry Value 15|Entry Value 16|Entry Value 17|Entry
Value 18|Entry Value 19|Entry Value 20|Entry Value 21|Entry Value 22|Entry Value 23|Entry Value 24|Entry
Value 25|Entry Value 26|Entry Value 27|Entry Value 28|Entry Value 29|Entry Value 30|Entry Value 31|Entry
Value 32|Entry Value 33|Entry Value 34|Entry Value 35|Entry Value 36|Entry Value 37|Entry Value 38|Entry
Value 39|Entry Value 40|Entry Value 41|Entry Value 42|Entry Value 43|Entry Value 44|Entry Value 45|Entry
Value 46|Entry Value 47|Entry Value 48|Entry Value 49|Entry Value 50|Segment 1|Segment 2|Segment 3|Segment
4|Segment 5|Segment 6|Segment 7|Segment 8|Segment 9|Segment 10|Segment 11|Segment 12|Segment 13|Segment 14|
Segment 15|Segment 16|Segment 17|Segment 18|Segment 19|Segment 20|Segment 21|Segment 22|Segment 23|Segment
24|Segment 25|Segment 26|Segment 27|Segment 28|Segment 29|Segment 30
```



Note: Entry Value 1 through 50 are based on the element. Similarly, Segment 1 through 30 are based on the cost structure.

Delete Element Entry

The Delete Element Entry task action uses the following columns to upload element entry values for the element specified in the workbook in the Element column. You can update multiple element entries for multiple elements per batch.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective As-of Date	Yes	DATE Date as of when to make this deletion effective. Must be in the format YYYY-MM-DD.
Element	Yes	VARCHAR2(80) Name of the existing element for the element entry to delete.
Entry Sequence	No	NUMBER The number representing the position of the element entry to delete as shown on the element entry summary page. When not specified, the default value is 1. For elements with multiple entry values, you must provide an entry value.
Person Number	No	NUMBER(18) A value for either this field, Assignment Number, Terms Number, or Relationship Existing person number of the person whose element entry want to create.

Column	Required	Comments
	Number is required. Don't specify a value here if you have a value for any of the other levels.	
Assignment Number	No A value for either this field, Person Number, Terms Number, or Relationship Number is required. Don't specify a value here if you have a value for any of the other levels.	NUMBER(18) Existing assignment number to create the element entry at the assignment level. Use this column when a person has multiple assignments for the same payroll relationship.
Terms Number	No A value for either this field, Person Number, Assignment Number, or Relationship Number is required. Don't specify a value here if you have a value for either of the other two levels.	NUMBER(18) Existing employment terms number to create the element entry at the employment terms level. Use this column when a person has multiple employment terms for the same payroll relationship.
Relationship Number	No A value for either this field, Person Number, Assignment Number, or Terms Number is required. Don't specify a value here if you have a value for any of the other levels.	NUMBER(18) Existing payroll relationship number to create the element entry at the payroll relationship level.
Payroll	No	VARCHAR2(30) Name of the payroll definition for the element entry to delete. Must be a valid payroll for the legislative data group used to create the batch. Payroll definition names are case-sensitive. Enter the name exactly.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Element Entry|Delete Element Entry||Effective As-of Date|*Element|Entry Sequence|*Person Number|Assignment Number|Terms Number|Relationship Number|Payroll

Related Topics

- [HCM Data Loader Process: Explained](#)
- [HCM Data Loader: Overview](#)

Elements Workbook

Import element details using the Batch Loader task in the Payroll Administration, Data Exchange, or Checklist work area. You enter the data in integrated Microsoft Excel workbook, save it to the staging tables, and then transfer the data into the

application. This topic explains the task actions you can add to the workbook for the Element task and the columns you complete for each task action.

This table explains the task actions that load element details.

Task Name	Purpose
Create Element Eligibility	Set up element eligibility for one or more elements.
Delete Element Eligibility	Delete element eligibility for one or more elements.
Create Input Value	Create input values and their validation details for one or more elements.
Update Input Value	Update input values and their validation details for one or more elements.
Delete Input Value	Delete input values from one or more elements.
Update Element	Update element details, such as entry dates and validation formulas for one or more elements.
Delete Element	End-date one or more elements.
Create People Group Key Flexfield Segment Combination	Create the people group flexfield segment combinations.
Delete People Group Key Flexfield Segment Combination	Delete the people group flexfield segment combinations.

Create Element Eligibility

The Create Element Eligibility task action uses the following columns to set up element eligibility.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective Date	Yes	DATE The first date the element eligibility is available for use. Must be in the format YYYY-MM-DD.
Element Eligibility Name	Yes	VARCHAR2(80) Name of the element eligibility record to create.
Element	Yes	VARCHAR2(80)

Column	Required	Comments
		Name of the existing element for the element eligibility to create.
All payrolls eligible	No	<p>VARCHAR2(30)</p> <p>Indicates whether any employee associated with a payroll can receive this element entry. This field appears as a check box on the Manage Elements page.</p> <p>Valid values are Yes and No.</p>
Payroll	No	<p>VARCHAR2(30)</p> <p>Name of the payroll definition for the element eligibility to create.</p> <p>Payroll definition names are case-sensitive. Enter the name exactly. If specified it applies to all employees that meet the criteria.</p>
Location	No	<p>NUMBER(18)</p> <p>If specified it applies to all employees that meet the criteria.</p>
Job	No	<p>NUMBER(18)</p> <p>If specified it applies to all employees that meet the criteria.</p>
Employment Category	No	<p>NUMBER(18)</p> <p>If specified it applies to all employees that meet the criteria.</p>
Relationship Type	No	NUMBER(22)
Grade	No	<p>VARCHAR2(30)</p> <p>Name of the grade that you created through the Manage Grades task.</p>
Position	No	Name of the position that you created through the Manage Positions task.
People Group Key Flexfield Structure	No	The people group key flexfield structure code that you created and attached to the enterprise.
People Group Segment Combination	No	The concatenated segments to which element criteria applies.

Column	Required	Comments
Department	No	TEXT(60) The name of the department that you created using the Manage Legal Entity task.
Payroll Statutory Unit	No	The name of the payroll statutory unit that you created using the Manage Legal Entity task.
Legal Employer	No	The name of the legal employer that you created using the Manage Legal Entity task.
Defaulting Formula	No	NUMBER(18) The ID of the defaulting formula.
Calculation Formula	No	NUMBER(18) The ID of the calculation formula.
Validation Formula	No	NUMBER(18) The ID of the validation formula.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

```
Element|Create Element Eligibility|*Effective Date|*Element Eligibility Name|*Element|All payrolls
eligible|Payroll|Location|Job|Employment Category|Relationship Type|Grade|Position|People Group Key
Flexfield Structure|People Group Segment Combination|Department|Payroll Statutory Unit|Legal Employer|
Defaulting Formula|Calculation Formula|Validation Formula
```

Delete Element Eligibility

The Delete Element Eligibility task action uses the following columns to delete element eligibility for one or more elements that you specify.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective Date	Yes	DATE The date as of when the element eligibility is deleted. Must be in the format YYYY-MM-DD.
Element Eligibility Name	Yes	VARCHAR2(80)

Column	Required	Comments
		Name of the element eligibility record to delete. To find the element eligibility name you can create a query from the Manage Element Definition, Element Object Navigator - Element Eligibility node or alternatively from the table element_link_name column-element_link_name.
Element	Yes	Name of the existing element for the element eligibility to delete.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Element|Delete Element Eligibility||*Effective Date|*Element Eligibility Name|*Element

Create Input Value

The Create Input Value task action uses the following columns to create input values and their validation details for one or more elements.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective As-of Date	Yes	DATE The first date the input value is available for use. Must be in the format YYYY-MM-DD.
Element	Yes	VARCHAR2(80) Name of the existing element for the input value to create.
Input Value	Yes	VARCHAR2(80) Name of the new input value.
Unit of Measure	Yes	VARCHAR2(30) Unit of measure for the input value, such as hours and monetary amount.
Required	Yes	VARCHAR2(1) Indicates whether entry for this input value is required.

Column	Required	Comments
		Valid values are Yes and No.
Display Sequence	Yes	NUMBER(5)
Default	No	VARCHAR2(60) Default value for the input value on entry. Value may be overridden for an element link, and for specific entries.
Apply default at runtime	Yes	VARCHAR2(1) Indicates if the input values will be defaulted at Run Time. Valid values are Yes and No.
Minimum	No	VARCHAR2(60) Minimum value allowed on entry. Can be overridden at the element eligibility level.
Maximum	No	VARCHAR2(60) Maximum value allowed on entry. Can be overridden at the element eligibility level.
Warning or Error	No	VARCHAR2(30) Indicates the kind of message that is generated if the input value is not invalid.
Lookup Type	No	VARCHAR2(30) List of allowable input values.
Validation Formula	No	NUMBER(13) ID of the formula used that generates the input value.
Validation Source	No	VARCHAR2(200)
Create a Database Item	Yes	VARCHAR2(1) Indicates whether a database item needs to be generated for this input value. Valid values are Yes and No.
Reference	No	VARCHAR2(80) Provides the balance context, such as state or county, which varies by country.

Column	Required	Comments
		You can see the list of valid values for a country in the Reference field on the Default Entry Values and Validation section on the Element Summary page.
Allow User Entry	Yes	<p>VARCHAR2(1)</p> <p>Indicates whether users can manually enter values for this input value.</p> <p>Valid values are Yes and No.</p>
Displayed	Yes	<p>VARCHAR2(30)</p> <p>Indicates whether the application displays this input value.</p> <p>Valid values are Yes (displayed) and No (hidden).</p>
Special Purpose	No	<p>Values come from a system lookup called PAY_RESERVED_INPUT_VALUE.</p> <p>Indicates if it holds a percentage value, a rate, or third-party payee details. This value assists with processing the input value based on what type of information it holds.</p>
Override Message	No	Not currently supported.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

```
Element|Create Input Value||*Effective As Of Date|*Element|*Input Value|*Unit Of Measure|*Required|*Display
Sequence|Default|*Apply default at run time|Minimum|Maximum|Warning Or Error|Lookup Type|Validation
Formula|Validation Source|*Create a Database Item|Reference|*Allow User Entry|*Displayed|Special Purpose|
Override Message
```

Update Input Value

The Update Input Value task action uses the following columns to update input values and their validation details for one or more elements.

Column	Required	Comments
Line Sequence	Yes	<p>NUMBER</p> <p>Enter 1 for the first row and continue sequentially for subsequent rows.</p>
Effective As-of Date	Yes	<p>DATE</p> <p>The date the update is available. Must be in the format YYYY-MM-DD.</p>

Column	Required	Comments
Element	Yes	<p>VARCHAR2(80)</p> <p>Name of the existing element for the input value to update.</p>
Input Value	Yes	<p>Name of the input value to update.</p>
Unit of Measure	No	<p>VARCHAR2(30)</p> <p>Unit of measure for the input value, such as hours and monetary units.</p>
Required	No	<p>VARCHAR2(1)</p> <p>Indicates whether entry for this input value is required.</p> <p>Valid values are Yes and No.</p>
Display Sequence	No	<p>NUMBER(5)</p>
Default	No	<p>VARCHAR2(60)</p> <p>Default value for the input value on entry. Value may be overridden for an element link, and for specific entries.</p>
Apply default at run time	No	<p>VARCHAR2(1)</p> <p>Indicates if the input values will be defaulted at Run Time.</p> <p>Valid values are Yes and No.</p>
Minimum	No	<p>VARCHAR2(60)</p> <p>Minimum value allowed on entry. Can be overridden at the element eligibility level.</p>
Maximum	No	<p>VARCHAR2(60)</p> <p>Maximum value allowed on entry. Can be overridden at the element eligibility level.</p>
Warning or Error	No	<p>VARCHAR2(30)</p> <p>Indicates the kind of message that is generated if the input value is not invalid.</p>
Lookup Type	No	<p>VARCHAR2(30)</p> <p>List of allowable input values.</p>

Column	Required	Comments
Validation Formula	No	NUMBER(13) ID of the formula used that generates the input value.
Validation Source	No	VARCHAR2(200)
Create a Database Item	Yes	VARCHAR2(1) Indicates whether the database item needs to be generated for this input value. Valid values are Yes and No.
Reference	No	VARCHAR2(80) Provides the balance context, such as state or county, which varies by country. You can see the list of valid values for a country in the Reference field on the Default Entry Values and Validation section on the Element Summary page.
Allow User Entry	Yes	VARCHAR2(1) Indicates whether users can manually enter values for this input value. Valid values are Yes and No.
Displayed	Yes	VARCHAR2(30) Indicates whether the application displays this input value.
Override Message	No	VARCHAR2(30) The ID of the validation override message used instead of a standard message when element entry input values are validated.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

```
Element|Update Input Value||*Effective As Of Date|*Element|*Input Value|Unit Of Measure|Required|Display
Sequence|Default|Apply default at run time|Minimum|Maximum|Warning Or Error|Lookup Type|Validation Formula|
Validation Source|Create a Database Item|Reference|Allow User Entry|Displayed|Override Message
```

Delete Input Value

The Update Input Value task action uses the following columns to delete input values for one or more elements.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective As-of Date	Yes	DATE The date the input value is no longer available. Must be in the format YYYY-MM-DD.
Element	Yes	VARCHAR2(80) Name of the existing element for the input value to delete.
Input Value	Yes	VARCHAR2(80) Name of the existing input value to delete.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Element|Delete Input Value||Effective As Of Date|*Element|*Input Value

Update Element

The Update Element task action uses the following columns to update element details, such as entry dates and validation formulas, for one or more elements.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective As-of Date	Yes	DATE The date the update is available. Must be in the format YYYY-MM-DD.
Element	Yes	VARCHAR2(80) Name of the existing element to update.
Earliest Entry Date	No	Corresponds to STARTING_ TIME_DEF_ID in pay_ element_ types_f. Valid values are from PAY_ TIME_ DEFINITIONS.


Column	Required	Comments
Latest Entry Date	No	Corresponds to ENDING_TIME_DEF_ID NUMBER(18) in pay_element_types_f. Valid values are from PAY_TIME_ DEFINTIONS.
Allow Multiple Entries in Same Period	No	VARCHAR2(1) Indicates whether a user can have multiple element entries for this element in the same payroll period. Valid values are Yes and No.
Additional Entry	No	VARCHAR2(1) Valid values are Yes and No.
Closed for Entry	No	VARCHAR2(1) Valid values are Yes and No.
Defaulting Formula	No	NUMBER(18) The ID of the defaulting formula.
Calculation Formula	No	NUMBER(18) The ID of the calculation formula.
Validation Formula	No	NUMBER(18) The ID of the validation formula.
Validation Override Message	No	VARCHAR2(30) The ID of the validation override message used instead of a standard message when element entry input values are validated.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

**Element|Update Element||*Effective As Of Date|*Element|Earliest Entry Date|Latest Entry Date|Allow multiple
entries in same period|Additional entry|Closed for entry|Defaulting Formula|Calculation Formula|Validation
Formula|Validation Override Message**

Delete Element

The Delete Element task action uses the following columns to end-date one or more elements.

 **Note:** The payroll batch loader only supports the end-dating of elements, not deleting elements.


Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective As-of Date	Yes	DATE The date the element is no longer available. Must be in the format YYYY-MM-DD.
Element	Yes	VARCHAR2(80) Name of the existing element to end-date.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Element|Delete Element||Effective As Of Date|*Element

Create People Group Key Flexfield Segment Combination


The Create People Group Key Flexfield Segment Combination task action uses the following columns.

 **Note:** This task creates people group flexfield segment combinations. The people group flexfield structure is configured manually through the Setup and Maintenance, Manage Key Flexfield task. This step is a prerequisite to creating element eligibility, which uses the people group flexfield combination.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Key Flexfield Structure	Yes	Name of the people group key flexfield structure to create.
Segment 1 - 30	Yes	Segment name as specified in the people group key flexfield.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Element|Create People Group Key Flexfield Segment Combination||*Key Flexfield Structure|Segment 1|Segment 2|Segment 3|Segment 4|Segment 5|Segment 6|Segment 7|Segment 8|Segment 9|Segment 10|Segment 11|Segment 12|Segment 13|Segment 14|Segment 15|Segment 16|Segment 17|Segment 18|Segment 19|Segment 20|Segment 21|Segment 22|Segment 23|Segment 24|Segment 25|Segment 26|Segment 27|Segment 28|Segment 29|Segment 30

 **Note:** Segment 1 through 30 are based on the cost structure.

Delete People Group Key Flexfield Segment Combination

The Delete People Group Key Flexfield Segment Combination task action uses the following columns.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Key Flexfield Structure	Yes	Name of the people group key flexfield structure to delete. Use the Manage Key Flexfields task to locate the key flexfield structure code that you configured.
People Group Segment Combination	Yes	Enter the concatenated segment combinations that need to be removed.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Element|Delete People Group Key Flexfield Segment Combination||*Key Flexfield Structure|*Segment Combination

Related Topics

- [Payroll Batch Loader Task Actions for Migrating Elements: Explained](#)

Globals Workbook

Import globals using the Batch Loader task in the Payroll Administration, Data Exchange, or Checklist work area. You enter the data in an integrated Microsoft Excel workbook, save it to the staging tables, and then transfer the data into the application. This topic explains the columns in the workbook for the Create task action for the Fast Formula Global task.

Globals are used to store values that are constant over a period of time and may be referenced in formulas, such as the name of a rate, a specific date, or a company terms.

Create Fast Formula Global

The Create task uses the following columns to create global values.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.

Column	Required	Comments
Effective Start Date	Yes	DATE Effective start date of the global value definition. Must be in the YYYY-MM-DD format.
Effective End Date	Yes	DATE Effective end date of the global value definition. Must be in the YYYY-MM-DD format.
Value	Yes	VARCHAR2(240) Value corresponding to the global that is defined by the customer.
Data Type	Yes	VARCHAR2(1) Type of data used to create the fast formula definition. Data types include date, number and string. Accepts values like N for Number and so on.
Name	Yes	VARCHAR2 Name of the global value definition. Definition names are case-sensitive. Ensure that you enter the name exactly.
Legislative Data Group	No	VARCHAR2(240) Name of the existing legislative data group used to create a global value definition. Legislative group names are case-sensitive. Make sure that you enter the name exactly.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Fast Formula Global|Create||*Effective Start Date|*Effective End Date|*Value|*Data Type|*Name|Legislative Data Group

Related Topics

- [Importing Data Using the Payroll Batch Loader: Explained](#)

Payment Methods Workbook

Import personal payment methods and payment methods for third-party organizations using the Batch Loader task in the Payroll Administration work area. You enter the data in integrated Microsoft Excel workbooks, save it to the staging tables,

and then transfer the data into the application. This topic explains the task actions you can add to the workbook for the Payment Method task and the columns for each task action.

This table explains the tasks that you can add to the workbook that load payment method information.

Task Name	Purpose
Create Personal Payment Method	Create personal payment methods details, such as allocation of EFT payments to a worker.
End-Date Personal Payment Method	End-date the payment method for the associated worker as of a specified date.
Delete Personal Payment Method	Permanently delete payment method details for the associated worker, including external bank accounts.
Create Third-Party Personal Payment Method	Associate a third-party person to a worker's assignment number, and optionally, an external bank account to receive EFT payments.
Delete Third-Party Personal Payment Method	Delete the association between the specified third-party person and a worker's assignment number including external bank account.
Update Third-Party Personal Payment Method	Update payment method details, such as the assignment number of the worker whose pay is subject to deduction and external bank account for EFT payments.
Create Third-Party Organization Payment Method	Associate a third-party organization payee to an organization payment method and optionally, an external bank account to receive EFT payments.
Delete Third-Party Organization Payment Method	Delete the association between a third-party organization and an external bank account.
Update Third-Party Organization Payment Method	Update payment method details, such as the organization payment method and the external bank account to receive EFT payments.


Create Personal Payment Method

The Create Personal Payment Methods task action uses the following columns to set up payment details for individual workers, such as allocations of EFT payments.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective Start Date	Yes	DATE The first date the payment method is available for use. Must be in the format YYYY-MM-DD.

Column	Required	Comments
Name	Yes	<p>VARCHAR2(250)</p> <p>Name of the payment method to create.</p>
Assignment Number	<p>No</p> <p>A value for either this field or Payroll Relationship Number is required.</p>	<p>NUMBER(18)</p> <p>Existing assignment number of the person whose payment information you want to create.</p> <p>Use this column when a person has multiple assignments for the same payroll relationship.</p>
Payroll Relationship Number	<p>No</p> <p>A value for either this field or Assignment Number is required.</p>	<p>NUMBER(18)</p> <p>Existing payroll relationship ID that identifies the person whose payment information you want to create.</p>
Amount	No	<p>NUMBER</p> <p>If you select Amount as the payment amount type, the amount value.</p>
Priority	Yes	<p>NUMBER(18)</p> <p>When multiple payment methods exist for a person, priority identifies which payment method to process first.</p>
Organization Payment Method	Yes	<p>VARCHAR2(80)</p> <p>Name of an existing organization payment method to use for payments to the specified third-party person. If you also provide bank account information, this value must be the name of an EFT method. The upload fails if you use the name of another method, such as check or cash.</p>
Percentage	No	<p>NUMBER(22)</p> <p>If you select Percentage as the payment amount type, the percentage value.</p>
Payment Amount Type	Yes	<p>VARCHAR2(30)</p> <p>Determines whether to use the Amount or Percentage columns to specify how much is paid. Valid values are Amount or Percentage.</p>
Bank Account Number	No	<p>NUMBER(18)</p> <p>The existing bank account number to receive EFT payments.</p>

Column	Required	Comments
--------	----------	----------

 **Tip:** Create bank account numbers using the Create External Bank Account task action.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payment Method|Create Personal Payment Method||*Effective Start Date|*Name|Assignment Number|Payroll Relationship Number|Amount|*Priority|*Organization Payment Method|Percentage|*Payment Amount Type|Bank Account Number

End-Date Personal Payment Method

The End-Date Personal Payment Methods task action uses the following columns to end personal payment methods for the specified assignment.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Assignment Number	Yes	NUMBER(18) Existing assignment number of the person whose payment method to end-date.
Name	Yes	VARCHAR2(250) Name of the payment method to end-date.
Effective As-of Date	Yes	DATE Date as of when to end the payment method. Must be in the format YYYY-MM-DD.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payment Method|End-Date Personal Payment Method||*Assignment Number|*Name|*Effective As-of Date

Delete Personal Payment Method

The Delete Personal Payment Methods task action uses the following columns to delete personal payment methods for individual workers.

Column	Required	Comments
Line Sequence	Yes	NUMBER

Column	Required	Comments
		Enter 1 for the first row and continue sequentially for subsequent rows.
Assignment Number	Yes	NUMBER(18) Existing assignment number of the person whose payment method to delete.
Name	Yes	VARCHAR2(250) Name of the payment method to delete.


If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payment Method|Delete Personal Payment Method||*Assignment Number|*Name

Create Third-Party Personal Payment Method

The Create Third-Party Personal Payment Methods task action uses the following columns to associate an existing third-party person to a worker's assignment number. You can optionally enter an existing external bank account to receive EFT payments.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective Date	Yes	DATE The first date the payment method is available for use. Must be in the format YYYY-MM-DD.
Party Name	Yes	VARCHAR2(250) Name of the existing third-party person for this payment method. Must be a valid name as stored in the HZ_PARTIES table.
Organization Payment Method	Yes	VARCHAR2(80) Name of an existing organization payment method to use for payments to the specified third-party person. If you also provide bank account information, this value must be the name of an EFT method. The upload fails if you use the name of another method, such as check or cash.
Assignment Number	Yes	NUMBER(18)

Column	Required	Comments
		Existing terms or assignment number of the worker whose pay is subject to deduction to pay the third-party person using this payment method.
Bank Account Number	No	NUMBER(18) The existing external bank account number to receive EFT payments.
 Tip: Create bank account numbers using the Create External Bank Account task action.		

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payment Method|Create Third-Party Personal Payment Method||*Effective Date|*Party Name|*Organization Payment Method|*Assignment Number|Bank Account Number

Delete Third-Party Personal Payment Method

The Delete Third-Party Personal Payment Methods task action uses the following columns to delete the payment method for a third-party person associated with a worker. This task action does not perform any validation and is intended for purging records as part of implementation or data migration.


Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Party Name	Yes	VARCHAR2(250) Name of the existing third-party person for this payment method. Must be a valid name as stored in the HZ_PARTIES table.
Assignment Number	Yes	NUMBER(18) Existing terms or assignment number of the worker whose pay is subject to deduction to pay the third-party person using this payment method.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payment Method|Delete Third-Party Personal Payment Method||*Party Name|*Assignment Number

Update Third-Party Personal Payment Method

The Update Third-Party Personal Payment Methods task action uses the following columns to update payment method details. For example, to update the assignment number of the associated worker whose pay is subject to deduction and external bank account to receive EFT payments.


Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Party Name	Yes	VARCHAR2(250) Name of the existing third-party person for this payment method. Must be a valid name as stored in the HZ_PARTIES table.
Organization Payment Method	Yes	VARCHAR2(80) Name of an existing organization payment method to use for payments to the specified third-party person. If you also provide bank account information, this value must be the name of an EFT method. The upload fails if you use the name of another method, such as check or cash.
Assignment Number	Yes	NUMBER(18) Existing terms or assignment number of the worker whose pay is subject to deduction using this payment method.
Bank Account Number	No	NUMBER(18) The existing bank account number to receive EFT payments.  Tip: Create bank account numbers using the Create External Bank Account task action.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payment Method|Update Third-Party Personal Payment Method|*Party Name|Organization Payment Method|*Assignment Number|Bank Account Number

Create Third-Party Organization Payment Method

The Create Third-Party Organization Payment Methods task action uses the following columns to associate the third party to an organization payment method. You can optionally enter an existing external bank account for EFT payments.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective Date	Yes	DATE The first date the payment method is available for use. Must be in the format YYYY-MM-DD.
Party Name	Yes	VARCHAR2(250) Name of the existing third-party organization for this payment method. Must be a valid name as stored in the HZ_PARTIES table.
Organization Payment Method	Yes	VARCHAR2(80) Name of an existing organization payment method to use for payments to the specified third-party person. If you also provide bank account information, this value must be the name of an EFT method. The upload fails if you use the name of another method, such as check or cash.
Bank Account Number	No	NUMBER(18) The existing bank account number to receive EFT payments.  Tip: Create bank account numbers using the Create External Bank Account task action.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payment Method|Create Third-Party Organization Payment Method|*Effective Date|*Party Name|*Organization Payment Method|Bank Account Number

Delete Third-Party Organization Payment Method

The Delete Third-Party Organization Payment Methods task action uses the following columns to delete third-party organization payment methods. This task action does not perform any validation and is intended for purging records as part of implementation or data migration.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Party Name	Yes	VARCHAR2(250) Name of the existing third-party organization for the payment method. Must be a valid name as stored in the HZ_PARTIES table.


If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payment Method|Delete Third-Party Organization Payment Method||*Party Name

Update Third-Party Organization Payment Method

The Update Third-Party Organization Payment Methods task action uses the following columns to update the bank account for EFT payments paid to the third-party organization.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Party Name	Yes	VARCHAR2(250) Name of the existing third-party organization for this payment method. Must be a valid name as stored in the HZ_PARTIES table.
Organization Payment Method	No	VARCHAR2(80) Name of an existing organization payment method to use for payments to the specified third-party person. If you also provide bank account information, this value must be the name of an EFT method. The upload fails if you use the name of another method, such as check or cash.
Bank Account Number	No	NUMBER(18)

Column	Required	Comments
		The existing bank account number to receive EFT payments.
		 Tip: Create bank account numbers using the Create External Bank Account task action.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payment Method|Update Third-Party Organization Payment Method||*Party Name|Organization Payment Method|Bank Account Number

Payroll Definitions Workbook

Import payroll definitions using the Batch Loader task in the Payroll Administration, Data Exchange, or Checklist work area. You enter the data in integrated Microsoft Excel workbooks, save it to the staging tables, and then transfer the data into the application. This topic explains the task actions you can add to the workbook for the Payroll Definition task and the columns you complete for each task action.

Using the payroll batch loader, you can create, update, or delete multiple payroll definitions for multiple legislative data groups at the same time. This table describes the batch upload tasks that manage payroll definitions.

Task Action	Purpose
Create Payroll Definition	Create a payroll definition and generate a schedule of payroll periods based on the specified dates, period type, and number of years.
Update Payroll Definition	Update an existing payroll definition, for example, to extend the number of years or change the associated ledger.
Delete Payroll Definition	Delete an unused payroll definition.

Create Payroll Definitions


Creating payroll definitions using the batch loader supports only dynamic date generation for payroll periods based on the calculation rules you specify in the workbook.

To specify calculation rules for dates:

1. Load rules for calculating the dates for payroll cycle events, such as date earned and payslip availability date.
2. For each date, you define the rules by specifying number of days (calendar or work days) before or after a base date.

Column	Required	Comments
Line Sequence	Yes	NUMBER

Column	Required	Comments
		Enter 1 for the first row and continue sequentially for subsequent rows.
Legislative Data Group	Yes	<p>VARCHAR2(240)</p> <p>Name of the existing legislative data group in which to create the payroll definition.</p> <p>Legislative group names are case-sensitive. Enter the name exactly.</p>
Effective As-of Date	Yes	<p>DATE</p> <p>The first date that processes can use the payroll definition for employee data. Use a date on or before the earliest date of any historical data you load.</p> <p>Must be in the format YYYY-MM-DD.</p>
Name	Yes	<p>VARCHAR2(80)</p> <p>Name of the payroll definition to create. Payroll definition names are case-sensitive. Enter the name exactly.</p>
Consolidation Group	Yes	<p>VARCHAR2(60)</p> <p>Name of the existing consolidation group in the specified legislative data group to use for the payroll definition.</p> <p>Consolidation group names are case-sensitive. Enter the name exactly.</p>
Period Type	Yes	<p>VARCHAR2(30)</p> <p>Frequency of the payroll cycle for the payroll definition. Valid values are Annually, Bimonthly, Biweekly, Monthly Calendar, Monthly Lunar, Quarterly, Semiannually, Semimonthly, and Weekly.</p> <p>Use the Manage Common Lookups task to see the lookup meanings for the PAY_PROC_ PERIOD_TYPE lookup type.</p> <p>Period types are case-sensitive. Enter the period type exactly.</p>
First Period End Date	Yes	<p>DATE</p> <p>Last date in the first payroll period during which the payroll is available for processing. Must be in the format YYYY-MM-DD.</p>

Column	Required	Comments
		 Tip: For semimonthly payroll definitions, the first period end date is typically the 15th of the month.
Number of Years	Yes	INTEGER(5) Number of years of time periods to generate starting from the beginning of the first period. Valid values are 1 through 10.
Fixed Date	Yes	VARCHAR2(80) Whether to generate the payroll calendar using fixed or dynamic offsets. Valid values are case-sensitive, either Yes or No. Use the Manage Common Lookups task to see the lookup meanings for the YES_NO lookup type.
Payroll Run Days	Yes	NUMBER(2) Number of days that the payroll run falls before or after the base date. Valid values are 0 through 99.
Date Earned Days	Yes	NUMBER(2) Number of days that date earned falls before or after the base date. Valid values are 0 through 99. The date earned must be within the effective dates of the payroll period.
Date Paid Days	Yes	NUMBER(2) Number of days that date paid falls before or after the base date. Valid values are 0 through 99.
Payslip Availability Days	Yes	NUMBER(2) Number of days that payslip availability falls before or after the base date. Valid values are 0 through 99.
Cutoff Days	Yes	NUMBER(2) Number of days that the cutoff date falls before or after the base date. Valid values are 0 through 99.
Payslip Available Before or After Base Date	Yes	VARCHAR2(1)

Column	Required	Comments
		<p>Indicates whether the payslip availability date should fall on, after, or before the base date for the payroll cycle event.</p> <p>Valid values are case-sensitive, either A (after), B (before), or O (on). Use the Manage Common Lookups task to see the lookup codes for the PAY_ DATE_OFFSET lookup type.</p>
Date Earned Before or After Base Date	Yes	<p>VARCHAR2(1)</p> <p>Indicates whether date earned should fall on, after, or before the base date for the payroll cycle event.</p> <p>Valid values are case-sensitive, either A (after), B (before), or O (on). Use the Manage Common Lookups task to see the lookup codes for the PAY_ DATE_OFFSET lookup type.</p>
Payroll Run Before or After Base Date	Yes	<p>VARCHAR2(1)</p> <p>Indicates whether the payroll run should fall on, after, or before the base date for the payroll cycle event.</p> <p>Valid values are case-sensitive, either A (after), B (before), or O (on). Use the Manage Common Lookups task to see the lookup codes for the PAY_ DATE_OFFSET lookup type.</p>
Cutoff Before or After Base Date	Yes	<p>VARCHAR2(1)</p> <p>Indicates whether the cutoff date should fall on, after, or before the base date for the payroll cycle event.</p> <p>Valid values are case-sensitive, either A (after), B (before), or O (on). Use the Manage Common Lookups task to see the lookup codes for the PAY_ DATE_OFFSET lookup type.</p>
Date Paid Before or After Base Date	Yes	<p>VARCHAR2(1)</p> <p>Indicates whether date paid should fall on, after, or before the base date for the payroll cycle event.</p> <p>Valid values are case-sensitive, either A (after), B (before), or O (on). Use the Manage Common Lookups task to see the lookup codes for the PAY_ DATE_OFFSET lookup type.</p>

Column	Required	Comments
Date Paid Day Type	Yes	<p>VARCHAR2(1)</p> <p>Type of days used to calculate payroll cycle event dates, such as calendar days or work days.</p> <p>Valid values are case-sensitive, either C (calendar days) or W (work days). Use the Manage Common Lookups task to see the lookup codes for the PAY_ BASE_OFFSET lookup type.</p>
Payslip Availability Day Type	Yes	<p>VARCHAR2(1)</p> <p>Type of days used to calculate payroll cycle event dates, such as calendar days or work days.</p> <p>Valid values are case-sensitive, either C (calendar days) or W (work days). Use the Manage Common Lookups task to see the lookup codes for the PAY_ BASE_OFFSET lookup type.</p>
Cutoff Day Type	Yes	<p>VARCHAR2(1)</p> <p>Type of days used to calculate payroll cycle event dates, such as calendar days or work days.</p> <p>Valid values are case-sensitive, either C (calendar days) or W (work days). Use the Manage Common Lookups task to see the lookup codes for the PAY_ BASE_OFFSET lookup type.</p>
Payroll Run Date Day type	Yes	<p>VARCHAR2(1)</p> <p>Type of days used to calculate payroll cycle event dates, such as calendar days or work days.</p> <p>Valid values are case-sensitive, either C (calendar days) or W (work days). Use the Manage Common Lookups task to see the lookup codes for the PAY_ BASE_OFFSET lookup type.</p>
Date Earned Day Type	Yes	<p>VARCHAR2(1)</p> <p>Type of days used to calculate payroll cycle event dates, such as calendar days or work days.</p> <p>Valid values are case-sensitive, either C (calendar days) or W (work days). Use the Manage Common Lookups task to see the lookup codes for the PAY_ BASE_OFFSET lookup type.</p>

Column	Required	Comments
Payslip Availability Base Date	Yes	<p>VARCHAR2(1)</p> <p>Date used as the basis for scheduling when workers can view their payslips.</p> <p>Valid values are case-sensitive, either E (period end date) or S (period start date). Use the Manage Common Lookups task to see the lookup codes for the PAY_BASE_DATE lookup type.</p>
Cutoff Base Date	Yes	<p>VARCHAR2(1)</p> <p>Date used as the basis for scheduling the final date for entering payroll information in the payroll period.</p> <p>Valid values are case-sensitive, either E (period end date) or S (period start date). Use the Manage Common Lookups task to see the lookup codes for the PAY_BASE_DATE lookup type.</p>
Payroll Run Date Base Date	Yes	<p>VARCHAR2(1)</p> <p>Date used as the basis for scheduling the date used by payroll calculation processes to retrieve effective values, such as employee details.</p> <p>Valid values are case-sensitive, either E (period end date) or S (period start date). Use the Manage Common Lookups task to see the lookup codes for the PAY_BASE_DATE lookup type.</p>
Date Earned Base Date	Yes	<p>VARCHAR2(1)</p> <p>Date used as the basis for scheduling the date on which the application processes element entries for the payroll run.</p> <p>Valid values are case-sensitive, either E (period end date) or S (period start date). Use the Manage Common Lookups task to see the lookup codes for the PAY_BASE_DATE lookup type.</p>
Date Paid Base Date	No	<p>VARCHAR2(1)</p> <p>Date used as a basis for scheduling the date the application marks employee records as paid.</p> <p>Valid values are case-sensitive, either E (period end date) or S (period start date). Use the Manage Common Lookups task to see</p>

Column	Required	Comments
		the lookup codes for the PAY_BASE_DATE lookup type.
Allow Negative Payments	Yes	<p>VARCHAR2(80)</p> <p>Whether the prepayments process can include negative payments.</p> <p>Valid values are case-sensitive, either Yes or No. Use the Manage Common Lookups task to see the lookup meanings for the YES_NO lookup type.</p>
Ledger	No	<p>VARCHAR2(30)</p> <p>Name of general ledger used to cost entries for the payroll run results in the legislative data group.</p> <p>Ledger names are case-sensitive. Enter the ledger name exactly to ensure the association with the ledger is correct. The batch upload and transfer processes don't perform validation on this value.</p>
Default Payment Method	No	<p>VARCHAR2(80)</p> <p>Name of a payment method already defined in the organization payment method to use for workers with no personal payment methods defined.</p>

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

```
Payroll Definition|Create Payroll Definition|*Legislative Data Group|*Effective As-of Date|*Name|
*Consolidation Group|*Period Type|*First Period End Date|*Number Of Years|*Fixed Date|*Payroll Run Days|
*Date Earned Days|*Date Paid Days|*Payslip Availability Days|*Cutoff Days|*Payslip Available Before or After
Base Date|*Date Earned Before or After Base Date|*Payroll Run Before or After Base Date|*Cutoff Before or
After Base Date |*Date Paid Before or After Base Date|*Date Paid Day Type|*Payslip Availability Day Type|
*Cutoff Day Type|*Payroll Run Date Day Type|*Date Earned Day Type|*Payslip Availability Base Date|*Cutoff
Base Date|*Payroll Run Date Base Date|*Date Earned Base Date|*Date Paid Base Date|*Allow Negative Payments|
Ledger|Default Payment Method
```

Update Payroll Definitions

The Update Payroll Definitions task workbook uses the following columns to update existing payroll definitions.

Column	Required	Comments
Line Sequence	Yes	<p>NUMBER</p> <p>Enter 1 for the first row and continue sequentially for subsequent rows.</p>
Legislative Data Group	Yes	<p>VARCHAR2(80)</p>


Column	Required	Comments
		Name of the existing legislative data group for the payroll definition to update.
Effective Start Date	Yes	DATE Effective start date of the payroll definition to update. Must be in the format YYYY-MM-DD.
Effective End Date	Yes	DATE The end date of the payroll definition to update. Must be in the format YYYY-MM-DD.
Name	Yes	VARCHAR2(30) Name of the payroll definition to update for the specified legislative data group. Payroll definition names are case-sensitive. Enter the name exactly.
Default Payment Method	No	VARCHAR2(80) Name of an existing organization payment method to use for workers with no personal payment methods defined.
Ledger	No	VARCHAR2(30) Name of the general ledger used to cost entries for the payroll run results in the specified legislative data group. Ledger names are case-sensitive. Enter the ledger name exactly.
Number of Years	No	INTEGER(5) Incremental number of years of time periods to generate starting from the beginning of the first payroll period. Valid values are 1 through 10.
Allow Negative Payments	Yes	VARCHAR2(80) Whether the prepayments process can include negative payments. Valid values are case-sensitive, either Yes or No. Use the Manage Common Lookups task to see the lookup meanings for the YES_NO lookup type.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

```
Payroll Definition|Update Payroll Definition||*Legislative Data Group|*Effective Start Date|*Effective End Date|*Name|Default Payment Method|Ledger|Number Of Years|*Allow Negative Payments
```

Delete Payroll Definitions

The Delete Payroll Definition task workbook uses the following columns.

 **Note:** You can't delete a payroll definition that is you associated with a person or that you used in a payroll run.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Legislative Data Group	Yes	VARCHAR2(80) Name of the existing legislative data group for the payroll definition to delete.
Effective Start Date	Yes	DATE Effective start date of the payroll definition to delete. Must be in the format YYYY-MM-DD.
Effective End Date	Yes	DATE End date of the payroll definition to delete. Must be in the format YYYY-MM-DD.
Name	Yes	VARCHAR2(80) Name of the existing payroll definition to delete for the specified legislative data group. Payroll definition names are case-sensitive. Enter the name exactly.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

```
Payroll Definition|Delete Payroll Definition||*Legislative Data Group|*Effective Start Date|*Effective End Date|*Name
```

Related Topics

- [Payroll Definitions: Explained](#)

Payroll Relationships Workbook

Import payroll assignment information using the Batch Loader task in the Payroll Administration, Data Exchange, or Checklist work area. You enter the data in integrated Microsoft Excel workbooks, save it to the staging tables, then transfer the data into the application. This topic explains the task actions you can add to the workbook for the Payroll Relationship task and the columns you complete for each task action.

This table explains the tasks that you can add to the workbook to manage the assigned payrolls for workers.

Task Action	Purpose
Add Payroll	Assign existing payrolls to employees
Delete Payroll	Remove assigned payrolls from employees
End Payroll	End-date assigned payrolls and update the element duration dates
Update Assigned Payroll Details	Update payroll relationship details at the assigned payroll level, such as whether a time card is required and overtime period
Update Assignment Details	Update payroll relationship details at the assignment level, such as whether a time card is required and overtime period
Update Payroll Element Duration Date	Overwrite values for specific time definitions for element duration dates of a payroll assignment

Add Payroll

The Add Payroll task action uses the following columns to assign existing payrolls to employees.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Assignment Number	Yes	NUMBER(18) The terms or assignment number of the employee.
Payroll	Yes	VARCHAR2(80) Name of the payroll to assign to the employee. Must be a valid payroll for the specified legislative data group or the

Column	Required	Comments
		legislative data group used to create the batch. Payroll definition names are case-sensitive. Enter the name exactly.
Effective As-of Date	Yes	DATE The first date the payroll is available for the employee. Must be on or before the earliest date of any historical data you load for the employee.
Legislative Data Group	No	VARCHAR2(240) The legislative data group of the payroll to assign. If not specified, the default value is legislative data group selected for the batch.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payroll Relationship|Add Payroll||*Assignment Number|*Payroll|*Effective As Of Date|Legislative Data Group

Delete Payroll

The Delete Payroll task action uses the following columns to remove assigned payrolls from a person's payroll relationship. This task action has no validation and is intended for use during integration testing or data migration to purge multiple assigned payrolls.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Assignment Number	Yes	NUMBER(18) The terms or assignment number of the employee.
Payroll	Yes	VARCHAR2(80) Name of the payroll to remove from the employee's payroll relationship. Must be a valid payroll for the specified legislative data group or the legislative data group used to create the batch. Payroll definition names are case-sensitive. Enter the name exactly.
Effective As-of Date	Yes	DATE

Column	Required	Comments
		The date as of when to remove the payroll assignment. Must be on or after the start date of the payroll assignment to remove and in the format YYYY-MM-DD.
Legislative Data Group	No	VARCHAR2(240) The legislative data group of the payroll to remove. If not specified, the default value is legislative data group selected for the batch.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payroll Relationship|Delete Payroll||*Assignment Number|*Payroll|*Effective As Of Date|Legislative Data Group

End Payroll

The End Payroll task workbook uses the following columns to end-date the assignment of payrolls to employees and enter element duration dates for ended assigned payroll records.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Assignment Number	Yes	NUMBER(18) The existing terms or assignment number of the employee.
Payroll	Yes	VARCHAR2(80) Name of the existing payroll in the payroll relationship to end-date. Must be a valid payroll for the specified legislative data group or the legislative data group used to create the batch. Payroll definition names are case-sensitive. Enter the name exactly.
Effective As-of Date	Yes	DATE The date as of when to end the assigned payroll.
Legislative Data Group	No	VARCHAR2(240) Name of the legislative data group of the payroll relationship to end-date. If not

Column	Required	Comments
		specified, the default value is legislative data group selected for the batch.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payroll Relationship|End Payroll||*Assignment Number|*Payroll|*Effective As Of Date|Legislative Data Group

Update Assigned Payroll Details

The Update Assigned Payroll Details task workbook uses the following columns to update payroll relationship details at the payroll level, such as whether a time card is required or the overtime premium for rate calculation.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective Date	Yes	DATE The date as of when this update is effective. Must be later than the effective date of the payroll definition and must be in the format YYYY-MM-DD.
Assignment Number	Yes	NUMBER(18) The existing terms or assignment number of the employee to update.
Payroll	Yes	VARCHAR2(80) Name of the existing payroll definition for the payroll relationship to update. Must be a valid payroll for the legislative data group used to create the batch. Payroll definition names are case-sensitive. Enter the name exactly.
Time Card Required	No	VARCHAR2(20) Whether time card entries are required before payment can occur. Value values are Yes or No. The default value is No.
Overtime Period	No	VARCHAR2(80) The name of the time definition to use for overtime premium rate calculation. The overtime premium rate applies to elements which have a secondary classification of Premium. If no value is provided, the payroll

Column	Required	Comments
		period will be used to calculate the premium rate.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payroll Relationship|Update Assigned Payroll Details||*Effective Date|*Assignment Number|*Payroll|Time Card Required|Overtime Period

Update Assignment Details

The Update Assignment Details task workbook uses the following columns to update payroll relationship details at the assignment level, such as whether a time card is required or the overtime premium for rate calculation.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective Date	Yes	DATE The date as of when this update is effective. Must be later than the effective date of the payroll definition and must be in the format YYYY-MM-DD.
Assignment Number	Yes	NUMBER(18) The existing terms or assignment number of the employee to update.
Payroll	Yes	VARCHAR2(80) Name of the existing payroll definition for the payroll relationship to update. Must be a valid payroll for the legislative data group used to create the batch. Payroll definition names are case-sensitive. Enter the name exactly.
Time Card Required	No	VARCHAR2(20) Whether time card entries are required before payment can occur. Value values are Yes or No. The default value is No.
Overtime Period	No	VARCHAR2(80) The name of the time definition to use for overtime premium rate calculation. The overtime premium rate applies to elements which have a secondary classification of

Column	Required	Comments
		Premium. If no value is provided, the payroll period will be used to calculate the premium rate.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Payroll Relationship|Update Assignment Details||*Effective Date|*Assignment Number|Time Card Required|Overtime Period

Update Payroll Element Duration Date

The Update Payroll Element Duration Date Columns task workbook uses the following columns to overwrite values for specific time definitions of element duration dates for an assigned payroll.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Effective Date	Yes	DATE The date as of when this update is effective. Must be on or after the start date of the payroll relationship and must be in the format YYYY-MM-DD.
Assignment Number	Yes	NUMBER(18) The existing terms or assignment number of the employee to update.
Payroll	Yes	VARCHAR2(80) Name of the existing payroll definition for the payroll relationship to update. Must be a valid payroll for the legislative data group used to create the batch. Payroll definition names are case-sensitive. Enter the name exactly.
Time Definition	Yes	VARCHAR2(80) The name of the time definition for the element duration value to update.
Date	Yes	DATE New date value to use for the specified time definition. Must be in the format YYYY-MM-DD.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:


```
Payroll Relationship|Update Payroll Element Duration Date||*Effective Date|*Assignment Number|*Payroll|*Time  
Definition|*Date
```

Third-Party Organizations Workbook

Import third-party organizations data using the Batch Loader task in the Payroll Administration, Data Exchange, or Checklist work area. You enter the data in an integrated Microsoft Excel workbook, save it to the staging tables, and then transfer the data into the application. This topic explains the task actions you can add to the workbook for the Third Party task and the columns you complete for each task action.


This table summarizes the three tasks that you can add to the workbook that third-party organization information.

Task Action	Purpose
Create Third-Party Organization	Create third-party organizations and their attributes.
Update Third-Party Organization	Update the D-U-N-S value of existing third-party organizations.
Merge Address	Add address data to third-party organizations.

 **Note:** Address data must exist prior to updating or creating third-party organizations using the batch loader tasks.

Create Third-Party Organization

The Create Third-Party Organization task workbook uses the following columns to create a registry ID, organization name, and party usage code.

 **Note:** You must the Merge Address task prior to creating third-party organizations using this task.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Registry ID	No	VARCHAR2(30) Unique ID to assign to the organization to create. Leave blank if the HZ_GENERATE_PARTY_NUMBER profile option in Cash Management is set to Auto numbering, update not allowed .

Column	Required	Comments
		Ensure that no organization with the same ID exists in your database.
Organization Name	Yes	VARCHAR2(240) Name of the organization to create.
Party Usage Code	Yes	VARCHAR2(120) Code to identify the organization as a trading community member. Valid values are: <ul style="list-style-type: none"> • External Payee • Payment Issuing Authority • Pension Provider • Professional Body • Bargaining Association • Disability Organization Valid party usage codes can be found in the lookup HZ_PARTY_USG_ASSIGNMENTS.
D-U-N-S Number	No	VARCHAR2(30) Unique numeric ID for the organization to create as assigned by Dun & Bradstreet Corporation.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Third Party|Create Third-Party Organization||*Registry ID|*Organization Name|*Party Usage Code|D-U-N-S Number

Update Third-Party Organization

The Update Third-Party Organization task action uses the following columns to update the D-U-N-S value of existing third-party organizations.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Registry ID	Yes	VARCHAR2(30) Unique registry ID of the existing organization to update.

Column	Required	Comments
Organization Name	No	<p>VARCHAR2(240)</p> <p>Name of the existing organization to update. Enter a name here to use if the organization can't be determined by the provided registry ID.</p> <p>Organization names are case-sensitive. Enter the name exactly.</p>
D-U-N-S Number	No	<p>VARCHAR2(30)</p> <p>Unique numeric ID for the existing organization to update as assigned by Dun & Bradstreet Corporation.</p>

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

Third Party|Update Third-Party Organization||*Registry ID|Organization Name|D-U-N-S Number

Merge Address

The Merge Address Columns task workbook uses the following columns to merge address data for the specified organization identified by its registry ID. After you create third-party organizations, you can use this task to upload a batch of address data for multiple organizations. You can also use this task to create third-party organizations along with their address data.

Column	Required	Comments
Line Sequence	Yes	<p>NUMBER</p> <p>Enter 1 for the first row and continue sequentially for subsequent rows.</p>
Registry ID	No	<p>VARCHAR2(30)</p> <p>ID of the existing organization to add address data to. If a record isn't found, a new party will be created.</p> <p>Leave blank for new organizations if the HZ_GENERATE_PARTY_NUMBER profile option in Cash Management is set to Auto numbering, update not allowed.</p>
Country	Required	<p>VARCHAR2(2)</p> <p>Country code. Must be a valid value from lookup FND_TERRITORIES_VL.</p>
Address Line 1	No	<p>VARCHAR2(240)</p>

Column	Required	Comments
		First line of address information.
Address Line 2	No	VARCHAR2(240) Second line of address information.
Address Line 3	No	VARCHAR2(240) Third line of address information.
Address Line 4	No	VARCHAR2(240) Fourth line of address information.
City	No	VARCHAR2(60) Town or city name.
County	No	VARCHAR2(60) County name.
Province	No	VARCHAR2(60) Province name.
State	No	VARCHAR2(60) State name.
Postal Code	No	VARCHAR2(60) National code to identify addresses in a specific country or territory.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

**Third Party|Merge Address||*Registry ID|Country|Address Line 1|Address Line 2|Address Line 3|Address Line 4|
City|County|Province|State|Postal Code**

Related Topics

- [Creating Third Parties: Points to Consider](#)


User-Defined Tables Workbooks

Import user-defined tables, columns, rows, and column instances using the HCM Data Loader. Alternatively, for small batches of data, you can use the Batch Loader task in the Payroll Administration, Data Exchange, or Check list work area. You enter the data in integrated Microsoft Excel workbooks, save it to the staging tables, and then transfer the data into the

application. This topic explains the task actions in the workbook for the HCM User-Defined Table task and the columns you complete for each task action.

This table explains the task actions that you can add to the workbook to create, update, and delete user-defined tables, columns, rows, and column instances.

Task Action	Purpose
Create User-Defined Table	Create a structure of rows and columns to maintain lists of values for use in the application.
Update User-Defined Table	Update the row title in a user-defined table.
Delete User-Defined Table	Permanently delete a structure of rows and columns to maintain lists of values in the application.
Create User-Defined Column	Create user-defined columns to store values for various types of operations. This action adds a column to an existing table.
Update User-Defined Column	Update the formula name for an existing user-defined column.
Delete User-Defined Column	Permanently delete user-defined columns in an existing user-defined table.
Create User-Defined Row	Create user-defined rows to maintain values that are stored in tables. This action adds rows to an existing table.
Update User-Defined Row	Update or correct user-defined rows to maintain values that are stored in tables.
Delete User-Defined Row	Delete user-defined rows in an existing user-defined table.
Create User-Defined Column Instance	Create user-defined instances (values) in user-defined tables.
Update User-Defined Column Instance	Update or correct user-defined instances of user-defined tables.
Delete User-Defined Column Instance	Delete user-defined column instances in an existing table.

 **Note:** User-defined tables, columns, rows, and column instances have interdependencies. If you are using the task actions for create or delete, you must use them in the proper sequence as shown in the following table. For example, before you can delete a user-defined table, you must first delete its column instances followed by its rows and columns.

Action	Sequence
Create	<ol style="list-style-type: none"> 1. Table 2. Row and column 3. Column instance
Delete	<ol style="list-style-type: none"> 1. Column instance 2. Row and column 3. Table

Create User-Defined Table

The Create User-Defined Table task action uses the following columns.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Legislative Data Group	Yes	VARCHAR2(240) Name of the existing legislative data group for the table to create. Legislative group names are case-sensitive. Ensure that you enter the name exactly.
User-Defined Row Title	No	VARCHAR2(80) Name to use as a title for a group of rows in the table.
User-Defined Table Name	Yes	VARCHAR2(80) Name of the user-defined table to create.
Range Or Match	Yes	VARCHAR2(30) Indicates the type of table to create. Valid values are: <ul style="list-style-type: none"> • R (for Range) • M (for Match) The lookup type table behind this row is FF_USER_TABLE_RANGE_OR_MATCH.
User Key Units	Yes	VARCHAR2(30) Indicates what data type are supported by the table. Valid values are: <ul style="list-style-type: none"> • D (date) • N (number) • T (text) For Range type tables, this value must be N. All data types are supported for Match type tables.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

HCM User-Defined Table|Create User-Defined Table||*Legislative Data Group|User-Defined Row Title|*User-Defined Table Name|*Range Or Match|*User Key Units

Update User-Defined Table

The Update User-Defined Table task action uses the following columns to update the row title in an existing user-defined table.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Legislative Data Group	Yes	VARCHAR2(240) Name of the existing legislative data group of the table to update. Legislative group names are case-sensitive. Ensure that you enter the name exactly.
User-Defined Table Name	Yes	VARCHAR2(80) Name of the existing user-defined table to update. Ensure that you enter the name exactly. The table must be a valid user-defined table for the specified legislative data group.
User-Defined Row Title	No	VARCHAR2(80) Name to use as a title for a group of rows in the specified table. If blank, no update occurs.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

HCM User-Defined Table|Update User-Defined Table||*Legislative Data Group|*User-Defined Table Name|User-Defined Row Title

Delete User-Defined Table

The Delete User-Defined Table task action uses the following columns to permanently delete a table.

 **Note:** You must perform delete operations in sequence starting with the child entities. Before you can delete a user-defined table, first delete its column instances followed by its rows and columns.

Column	Required	Comments
Line Sequence	Yes	NUMBER

Column	Required	Comments
		Enter 1 for the first row and continue sequentially for subsequent rows.
Legislative Data Group	Yes	VARCHAR2(240) Name of the existing legislative data group used for the user-defined table to delete. Legislative group names are case-sensitive. Ensure that you enter the name exactly.
User-Defined Table Name	Yes	VARCHAR2(80) Name of the user-defined table to delete. Ensure that you enter the name exactly. The table must be a valid user-defined table defined for the specified legislative data group.
User-Defined Row Title	No	Leave blank.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

HCM User-Defined Table|Delete User-Defined Table||*Legislative Data Group|*User-Defined Table Name|User-Defined Row Title

Create User-Defined Column

The Create User-Defined Column task action uses the following columns to create columns in an existing user-defined table.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Legislative Data Group	Yes	VARCHAR2(240) Name of the existing legislative data group of the user-defined table for the column to create. Legislative group names are case-sensitive. Ensure that you enter the name exactly.
User-Defined Table Name	Yes	VARCHAR2(80) The name of the existing user-defined table for the column to create.
Data Type	Yes	VARCHAR2(1)

Column	Required	Comments
		<p>The type of data for the user-defined column. Valid values are:</p> <ul style="list-style-type: none"> • D (date) • N (number) • T (text) <p>For Range type tables, this value must be N. All data types are supported for Match type tables.</p>
Formula Effective Date	No	Leave blank.
Formula Name	No	<p>VARCHAR2(80)</p> <p>The name of valid and compiled User Table Validation formula.</p>
User-Defined Column Name	Yes	<p>VARCHAR2(80)</p> <p>The name of the column to create.</p>

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

HCM User-Defined Table|Create User-Defined Column|*Legislative Data Group|*User-Defined Table Name|Data Type|Formula Effective Date|Formula Name|*User-Defined Column Name

Update User-Defined Column

The Update User-Defined Column task action uses the following columns to update the formula name for an existing column.

Column	Required	Comments
Line Sequence	Yes	<p>NUMBER</p> <p>Enter 1 for the first row and continue sequentially for subsequent rows.</p>
Legislative Data Group	Yes	<p>VARCHAR2(240)</p> <p>Name of the existing legislative data group of the user-defined table whose column to update. Legislative group names are case-sensitive. Ensure that you enter the name exactly.</p>
User-Defined Table Name	Yes	<p>VARCHAR2(80)</p> <p>The name of the user-defined table whose columns to update. Ensure that you enter the name exactly. The table must be a valid user-defined table for the specified legislative data group.</p>

Column	Required	Comments
User-Defined Column Name	Yes	VARCHAR2(80) Name of the user-defined column for which you want to update the record. Ensure that you enter the name exactly.
Formula Effective Date	No	Leave blank.
Formula Name	No	VARCHAR2(80) The name of valid and compiled User Table Validation formula. The formula effective date is taken into consideration when the column is updated. If this column is blank, no update is made.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

HCM User-Defined Table|Update User-Defined Column||*Legislative Data Group|*User-Defined Table Name|*User-Defined Column Name|Formula Effective Date|Formula Name

Delete User-Defined Column

The Delete User-Defined Column task action uses the following columns to permanently delete user-defined columns in an existing user-defined table.

 **Note:** Before you delete a user-defined column, first delete its column instances.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Legislative Data Group	Yes	VARCHAR2(240) Name of the existing legislative data group of the user-defined table whose column to delete. Legislative group names are case-sensitive. Ensure that you enter the name exactly.
User-Defined Table Name	Yes	VARCHAR2(80) The name of the existing user-defined table whose column to delete. Ensure that you enter the name exactly. The table must be a valid user-defined table for the specified legislative data group.

Column	Required	Comments
User-Defined Column Name	Yes	VARCHAR2(80) Name of the user-defined column to delete. Must be a valid column for the specified user-defined table. Ensure that you enter the name exactly.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

HCM User-Defined Table|Delete User-Defined Column||*Legislative Data Group|*User-Defined Table Name|*User-Defined Column Name

Create User-Defined Row

The Create User-Defined Row task action uses the following columns to add rows to existing user-defined tables.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Match Value Or Low Range Value	No	VARCHAR2(80) If the type of user-defined table is Range, the value you enter must be a number representing the lowest range of the row. If the type of user-defined table is Match, enter the value to match.
Legislative Data Group	Yes	VARCHAR2(240) Name of the existing legislative data group of the user-defined table whose row to create. Legislative group names are case-sensitive. Ensure that you enter the name exactly.
User-Defined Table Name	Yes	VARCHAR2(80) Name of the existing user-defined table whose rows to create. Ensure that you enter the name exactly. The table must be a valid user-defined table defined for the specified legislative data group.
High Range	No	VARCHAR2(80) High range of the row in cases where the type of user-defined table is Range. For Match type tables, leave this column blank.
Sequence	No	NUMBER(5)


Column	Required	Comments
		Number that indicates where the row appears sequentially in the table.
Row Name	Yes	<p>VARCHAR2(80)</p> <p>Name of the row to create. The row name must be unique for a user-defined table.</p> <p>The row name must be equal to the row low range or name during creation of the user-defined row to maintain uniqueness of the row.</p>
Effective Start Date	Yes	<p>DATE</p> <p>Effective start date of the row. Must be in the format YYYY-MM-DD.</p>
Effective End Date	Yes	<p>DATE</p> <p>Effective end date of the row. Must be in the format YYYY-MM-DD.</p>

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

HCM User-Defined Table|Create User-Defined Row||Match Value Or Low Range Value|*Legislative Data Group|*User-Defined Table Name|High Range|Sequence|*Row Name|*Effective Start Date|*Effective End Date

Update User-Defined Row

The Update User-Defined Row task action uses the following columns to update or correct rows in an existing user-defined table.

 **Note:** The effective start and end dates for task action are part of the unique identifier for user-defined rows. You must provide the effective dates exactly as they were originally when creating the row.

Column	Required	Comments
Line Sequence	Yes	<p>NUMBER</p> <p>Enter 1 for the first row and continue sequentially for subsequent rows.</p>
Legislative Data Group	Yes	<p>VARCHAR2(240)</p> <p>Name of the existing legislative data group of the user-defined table whose rows to update. Legislative group names are case-sensitive. Ensure that you enter the name exactly.</p>
User-Defined Table Name	Yes	<p>VARCHAR2(80)</p>

Column	Required	Comments
		Name of the existing user-defined table whose rows to update. Ensure that you enter the name exactly. The table must be a valid user-defined table defined for the specified legislative data group.
Row Name	Yes	VARCHAR2(80) Name of the existing row to update. Must be valid for the specified table. Ensure that you enter the name exactly.
Effective Start Date	Yes	DATE Effective start date of the row to update in the format YYYY-MM-DD.
Effective End Date	Yes	DATE Effective end date of the row to update in the format YYYY-MM-DD.
Match Value Or Low Range Value	No	VARCHAR2(80) Enter a new low range or match value. If there is no update for this column, enter the existing low range or match value.
High Range	No	VARCHAR2(80) For Range tables, either enter a new high range value. If there is no update for this column, enter the existing high range value.
Sequence	No	NUMBER Enter a new sequence number to indicate where the row must appear sequentially in the table. If there is no update for this column, enter the existing sequence number.
Date-Effective Operation	No	Leave blank.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

```

HCM User-Defined Table|Update User-Defined Row||*Legislative Data Group|*User-Defined Table Name|*Row Name|
*Effective Start Date|*Effective End Date|Match Value Or Low Range Value|High Range|Sequence|Date-Effective
Operation

```

Delete User-Defined Row

The Delete User-Defined Row task action uses the following columns to delete rows in user-defined tables.

 **Note:** Before you delete a user-defined row, you must first delete its column instances.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Legislative Data Group	Yes	VARCHAR2(240) Name of the existing legislative data group of the user-defined table whose row to delete. Legislative group names are case-sensitive. Ensure that you enter the name exactly.
User-Defined Table Name	Yes	VARCHAR2(80) Name of the existing user-defined table whose rows to delete. Ensure that you enter the name exactly. The table must be a valid user-defined table defined for the specified legislative data group.
Row Name	Yes	VARCHAR2(80) Name of the existing user-defined row to delete. Ensure that you enter the name exactly.
Effective Start Date	Yes	DATE Effective start date of the row to delete. Must be on or after the effective start date of the row to delete and must be in the format YYYY-MM-DD.
Effective End Date	Yes	DATE Date on which to delete the row. Must on or before the effective end date of the row to delete and must be in the format YYYY-MM-DD.
Match Value Or Low Range Value	No	Leave blank.
High Range	No	Leave blank.
Sequence	No	Leave blank.
Date-Effective Operation	No	Leave blank.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

```

HCM User-Defined Table|Delete User-Defined Row||*Legislative Data Group|*User-Defined Table Name|*Row Name|
*Effective Start Date|*Effective End Date|Match Value Or Low Range Value|High Range|Sequence|Date-Effective
Operation

```

Create User-Defined Column Instance

The Create User-Defined Column Instance task action uses the following columns to create column instances for an existing user-defined table.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Legislative Data Group	Yes	VARCHAR2(240) Name of the existing legislative data group of the user-defined table, row, and column for the column instance to create. Legislative group names are case-sensitive. Ensure that you enter the name exactly.
User-Defined Table Name	Yes	VARCHAR2(80) Name of the existing user-defined table for the column instance to create. Ensure that you enter the name exactly. The name must be a valid user-defined table that is defined for the specified legislative data group.
Row Name	Yes	VARCHAR2(80) Name of the existing row in the specified user-defined table for the column instance to create. Ensure that you enter the name exactly. The row name must be a valid row in the specified user-defined table.
User-Defined Column Name	Yes	VARCHAR2(80) Name of the existing column for the column instance to create. The column name must be a valid column in the specified user-defined table. Ensure that you enter the name exactly.
Effective Start Date	Yes	DATE Effective date of the user-defined column instance to create. Must be a date within the span of the corresponding user-defined row, and must be in the format YYYY-MM-DD.

Column	Required	Comments
Effective End Date	Yes	DATE End date of the column instance to create. Must be a date within the span of the corresponding user-defined row, and must be in the format YYYY-MM-DD.
Value	Yes	VARCHAR2(255) Value that can be referenced in a formula or in the user interface.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

HCM User-Defined Table|Create User-Defined Column Instance||*Legislative Data Group|*User-Defined Table Name|*Row Name|*User-Defined Column Name|*Effective Start Date|*Effective End Date|*Value

Update User-Defined Column Instance

The Update User-Defined Column Instance task action uses the following columns to update or correct user-defined column instances.

Column	Required	Comments
Line Sequence	Yes	NUMBER Enter 1 for the first row and continue sequentially for subsequent rows.
Legislative Data Group	Yes	VARCHAR2(240) Name of the existing legislative data group of the user-defined table for the column instance to update. Legislative group names are case-sensitive. Ensure that you enter the name exactly.
User-Defined Table Name	Yes	VARCHAR2(80) Name of the existing user-defined table of the column instance to update. Ensure that you enter the name exactly. The name must be a valid user-defined table that is defined for the specified legislative data group
User-Defined Column Name	Yes	VARCHAR2(80) The existing user-defined column name of the column instance to update. The name must be a valid column in the specified user-defined table.

Column	Required	Comments
Row Name	Yes	<p>VARCHAR2(80)</p> <p>Name of the existing row of the column instance to update. Ensure that you enter the name exactly. The name must be a valid row in the specified user-defined table.</p>
Effective Start Date	Yes	<p>DATE</p> <p>Effective start date of the column instance to update. To correct the column instance, this value must match the effective start date of the corresponding row. To update the column instance enter a date on or after the effective start date of the corresponding row. Must be in the format YYYY-MM-DD.</p>
Effective End Date	Yes	<p>DATE</p> <p>Effective end date of the column instance to update. To correct the column instance, this value must match the end date of the corresponding row. To update the column instance, this value must be on or after the end date of the corresponding row. Must be in the format YYYY-MM-DD.</p>
Value	No	<p>VARCHAR2(255)</p> <p>New value that can be referenced in a formula or in the user interface.</p>
Date-Effective Operation	No	<p>Leave blank.</p>

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

HCM User-Defined Table|Update User-Defined Column Instance||*Legislative Data Group|*User-Defined Table Name|*User-Defined Column Name|*Row Name|*Effective Start Date|*Effective End Date|Date-Effective Operation|Value

Delete User-Defined Column Instance

The Update User-Defined Column Instance task action uses the following columns to delete existing column instances.

Column	Required	Comments
Line Sequence	Yes	<p>NUMBER</p> <p>Enter 1 for the first row and continue sequentially for subsequent rows.</p>
Legislative Data Group	Yes	<p>VARCHAR2(240)</p>

Column	Required	Comments
		Name of the existing legislative data group of the user-defined table for the column instance to delete. Legislative group names are case-sensitive. Ensure that you enter the name exactly.
User-Defined Table Name	Yes	VARCHAR2(80) Name of the existing user-defined table of the column instance to delete. Ensure that you enter the name exactly. The name must be a valid user-defined table that is defined for the specified legislative data group
User-Defined Column Name	Yes	VARCHAR2(80) The existing user-defined column name of the column instance to delete. The name must be a valid column in the specified user-defined table.
Row Name	Yes	VARCHAR2(80) Name of the existing row of the column instance to delete. Ensure that you enter the name exactly. The name must be a valid row in the specified user-defined table.
Effective Start Date	Yes	DATE Effective start date of the column instance to delete. This date must be on or after the effective start date of the column instance to delete and must be in the format YYYY-MM-DD.
Effective End Date	Yes	DATE Date on which to delete the column instance. Must be on or before the effective end date of the column instance to delete and must be in the format YYYY-MM-DD.
Date-Effective Operation	No	Leave blank.

If you are uploading a file for this task instead of entering data directly into the workbook, you must use the following format:

HCM User-Defined Table|Delete User-Defined Column Instance||*Legislative Data Group|*User-Defined Table Name|*User-Defined Column Name|*Row Name|*Effective Start Date|*Effective End Date|Date-Effective Operation

Related Topics

- [Importing Data Using the Payroll Batch Loader: Explained](#)
- [User Table Validation Formula Type](#)

- HCM Data Loader Process: Explained
- HCM Data Loader: Overview

24 Time, Absence, and Pension Data for Payroll

Overview

You can automate the regular import of time card entries, absence entries, and pension enrollments. You use predefined flows where you specify the interface type and XML file containing the data to load, unless the application pushes the data directly to payroll. For example, recording an absence in Oracle Fusion Absence Management automatically transfers the data to payroll.

This topic covers:

- Loading data options
- Using an XML file to load data

If you're loading data to element entries and not to calculation cards, you can use the payroll batch loader.

Loading Data Options

The application you use to record time, absences, and pensions determines options for importing data to payroll, as shown in the following table.

Source Application	Data Load Options
Time card	<ul style="list-style-type: none">• For Oracle Fusion Time and Labor, submit the Load Time Card Batches process.• For a third-party time application, submit the Load Time Card Batches process and load an XML file.
Absences	<ul style="list-style-type: none">• For Absence Management, use the Manage Absence Records task to record an absence and transfer it directly to the person's absence calculation card. When you set up the plan in Absence Management, you must select the Transfer absence payment information for payroll processing check box. Find this setting in the Payroll Integration section of the Entries and Balances tab on the Create Absence Plan page.• For third-party absence applications, submit the Load Absence Batches process and load an XML file.
Pensions	<ul style="list-style-type: none">• Submit the Load Benefit Batches process and load an XML file. The availability of this process depends on the implementation for your country or territory.

Using an XML File to Load Data

To load data by submitting a process to transfer data using an XML file format, do the following steps:

1. Extract your data into an XML file, using the required XML file format and tags.

For time card entries, if your file is over 2 MB, stage the file on the Oracle WebCenter Content Server and note the Content Id.

2. Select the Submit a Payroll Flow task from the Payroll Checklist work area and select the appropriate process: Load Time Card Batches, Load Absences Batches, or Load Benefits Batches.
3. On the Enter Parameters page, enter a name to identify the flow.
4. In the Interface field, select the option to import an XML file.
5. For time card entries, if your file is over 2 MB, specify the number for the **Content ID** submission parameter.
6. Optionally, specify a process configuration group.
7. Optionally, schedule the process to transfer the time card entries, absences, or benefit enrollments to payroll.
8. Click the **Submit** button on the Review page. In the Confirmation window, click the **Ok and View Checklist** button.
9. On the Task Details tab of the payroll flow, click the **Go To Task** button for the Upload File task.

or time card entries, if your file is over 2 MB, skip the Upload File task, and the load batches tasks starts automatically.

10. On the Upload File page in the File field, click the **Add** button to display the Manage Attachments window.
11. In the Type field, select **File**.
12. Search for and select your XML file. Click the **OK** button.

The load batches task starts automatically after the Upload File task completes.

When the process completes successfully, it creates or updates a calculation card for each person included in the XML file. Use the Manage Calculation Cards task in the Payroll Calculation work area to view the new entries on the cards.

Related Topics

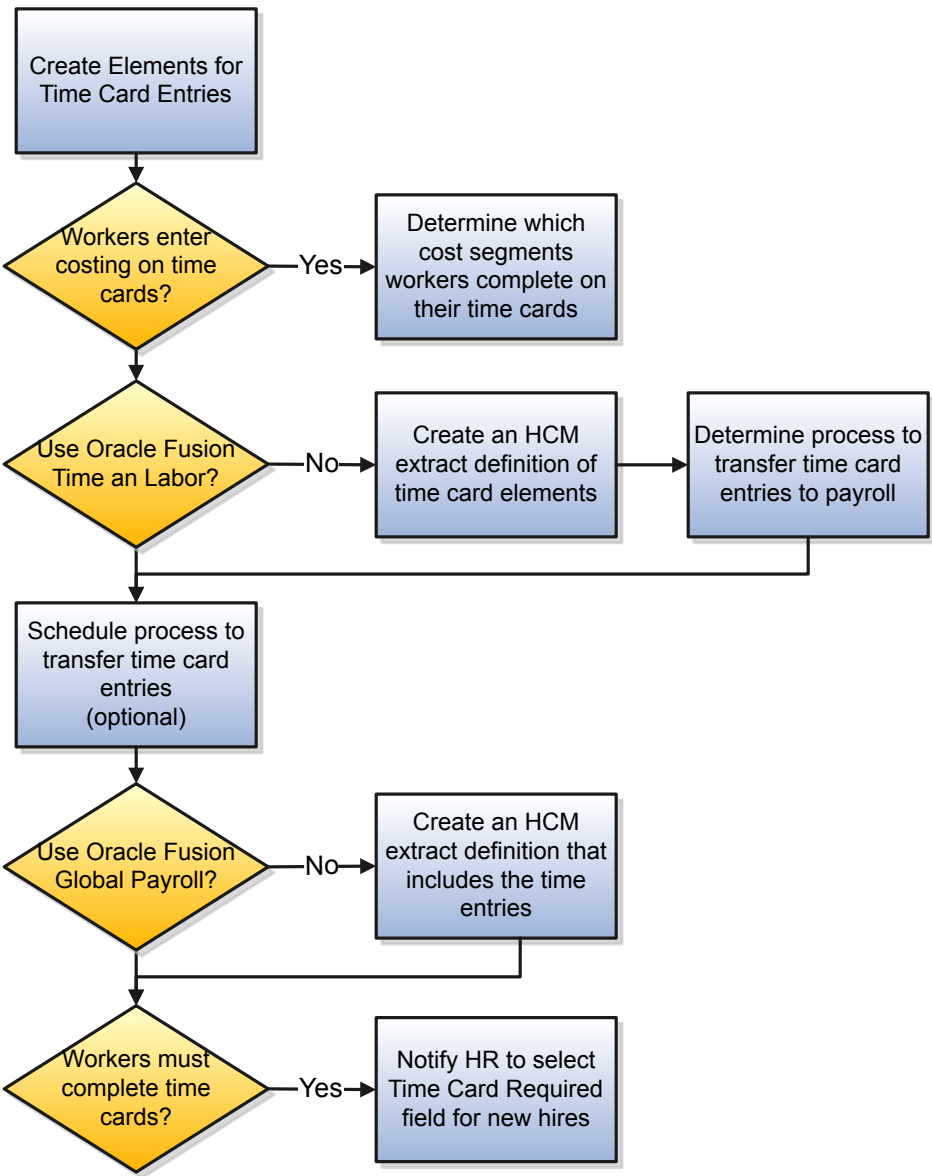
- [Creating Element Entries Using the Batch Loader: Worked Example](#)

Prerequisite Payroll Setup for Importing Time Entries: Explained

Before you can import time card entries for processing in a payroll run or extracting for a third-party payroll provider, you complete payroll setup tasks. These tasks vary depending on your configuration, specifically whether your enterprise:


- Uses Oracle Fusion Time and Labor or a third-party time provider
- Uses Oracle Fusion Global Payroll or a third-party payroll provider
- Requires workers to submit a time card if their pay is calculated using time entries

The following figure and table lists the tasks that depend on each of these configuration choices.



All configurations create elements for use with time cards. The following table describes the remaining setup tasks and processes that vary based on your configuration.

Task	Applies To
Create elements for time card entries	All configurations
Determine which cost segments workers complete on their time cards	Optional for all configurations

Task	Applies To
Record costing overrides at the element entry level by having workers specify the account to charge on their time cards	
 Note: If you enable costing overrides, confirm you are using the segments that are enabled for element entry level costing in the Cost Allocation key flexfield.	
Create an HCM extract definition of time card elements	Third-party time provider
Determine process to transfer time card entries to payroll:	Third-party time provider
<ul style="list-style-type: none"> • Use the Load Time Card Batches process to upload time entries in an XML file to calculation cards • Use the Payroll Batch Loader to load element entries 	
Schedule process to transfer time card entries	Optional for all configurations
As an example, schedule the Load Time Card Batches process:	
<ul style="list-style-type: none"> • After normal working hours to distribute the load on server • To run more frequently during time periods when most employees submit their time cards 	
Create an HCM extract definition that includes time entries	Third-party payroll provider
Notify HR to select Time Card Required field at the Terms or Assignment level when a new hire's pay calculations will depend on time cards	Optional for all configurations

Related Topics

- [Scheduling Flows: Explained](#)
- [Extract Components: How They Work Together](#)
- [Time Card Required Option: Critical Choices](#)

Creating Elements for Time Card Entries: Procedure

You create nonrecurring elements to process pay based on time card entries, such as elements for regular, overtime, double-time, and shift pay. Creating a time card element generates all the related elements, balances, formulas, and calculation components. You then transfer the elements to your time provider.

This topic covers the following procedures:

- Creating earnings elements
- Creating calculation components for standard-category elements
- Converting elements for use in time cards
- Setting up area overrides
- Setting up costing overrides

Creating Earnings Elements

The steps for creating a time card element depend on whether the time card template is available for your country. If the template is available, follow the steps in this section. Otherwise, create an earnings element using the Standard category, and specify an hours multiplied by rate calculation rule. Follow the steps in the section below to create calculation components for your element.

Complete these steps to create an element using the time card template:


1. Create an earnings element on the Manage Elements page of the Payroll Calculation work area.
2. Select a primary classification of standard or supplemental earnings.
3. Select the Time Card category.
4. Complete the information on the Basic Details page.
5. On the Additional Details page:
 - a. Select the calculation units to use in reports.

Typically, you select time units that match the time units entered on time cards for that element. If you select different units, the application uses 8 hours to convert days to hours.

- b. Optionally, select a default rate to calculate time.

When calculating the run result for the element entry, the formula uses the default rate unless a rate is entered on the person's time card.

6. Complete the element eligibility information for the new time element, and its associated retroactive and related elements, such as the result and calculation elements.

 **Note:** If the straight time portion of overtime is reported separately from regular time, create two elements, such as overtime and overtime premium elements. If the regular and straight time portions of overtime are reported together, you might use straight time instead of regular time, and create a separate element for the overtime premium.

Creating Calculation Components for Standard-Category Elements


You can create calculation components for elements created with the Standard category rather than the Time Card category. Complete the following steps for each existing element:

1. Submit the Create Time Card Calculation Components process from the Payroll Checklist or Payroll Administration work area.

These elements must have a calculation rule of hours multiplied by rate.
2. Complete the element eligibility information for the element and its associated retroactive and its related elements, including the result element, and the element with a suffix of CIR.
3. After you run the process to convert your elements, submit the Compile Formula process in the Manage Payroll Calculations work area. Perform a bulk compile by entering wild cards in the Formula and Formula Type parameters.

Generate Time Card Fields for Your Elements

After creating elements, generate time card fields for them. For Time and Labor, perform the processes listed in the following table using the Define Time and Labor task list in the Setup and Maintenance work area.

Process	Description
Generate Data Dictionary Time Attributes	<p>Creates dependent payroll attributes for all element input values, such as hours and rate.</p> <p>You must run the Generate Data Dictionary Time Attributes process after making any changes to time elements. Such changes include adding or deleting elements, editing input values, or editing element eligibility records.</p> <div> Caution: Failure to run the process might negatively impact the setup of time card fields, the validation of payroll time types, or the transfer of time to payroll.</div>
Generate Time Card Fields	<p>Creates time card fields using the data dictionary time attributions for the specified legislative data group.</p> <p>You can use the Manage Time Card Fields task to create time card fields for single and multiple attributes.</p>

If you are using a third-party time provider, create an HCM extract for the time card elements. The extract includes the element's mapping ID that you specify in the XML file when you transfer the time entries to payroll.

Setting Up Area Overrides

Some countries or territories create time card elements with area input values for use as overrides. The overrides enable employers to tax employees based on where they work. For example, employees would specify the area information where they worked on temporary assignment while away from their normal work location. These area entries are then included in the time card records transferred to payroll by the Load Time Card Batches process.

Setting Up Costing Overrides

You can specify additional attributes in Time and Labor to enter costing segment values on time cards. The segments must match the segments that you can enter on element entries. Use the Manage Element Entries task in the Payroll Calculation

and search for a person's record. View the available segments on the Costing tab of the Manage Person Details page. Costing is defined on the element eligibility record of the results element. When you transfer time entries, the transfer process displays the costing on the calculation element.

As an example, the structure of your cost allocation key flexfield might specify that the department segment is entered at the element entry level. You could specify this additional attribute on the time card. Your employees could then specify the department to charge for overtime hours worked while on loan to a different department. After you transfer the time entries, the payroll calculation uses the department specified for the overtime hours to derive the costing results.

Use the Manage Element Entries task in the Payroll Calculation work area to view costing overrides transferred to payroll. The results display on the Costing tab of the Manage Person Details page.

Related Topics

- [Processing Time Entries in Payroll: Explained](#)
- [Time Card Required Option: Critical Choices](#)
- [Elements: How They Hold Payroll Information for Multiple Features](#)

Importing Time Card Entries to Payroll: Procedure

Import time card entries to payroll by submitting the Load Time Card Batches process from the Payroll Checklist or Payroll Administration work areas. The process creates a new calculation card or updates an existing card for each person with time entries included in the batch.


You use the same process to import time entries from Oracle Fusion Time and Labor and from a third-party time provider. Submitting the time entries for a third-party requires an additional step of creating an XML file that contains the time entries. This topic covers both procedures:

- Importing time entries from Time and Labor
- Importing time entries from a third-party time provider


Importing Time Entries from Time and Labor

Complete the following steps to import your time entries

1. Use the Submit a Payroll Flow task to submit the Load Time Card Batches flow from the Payroll Administration or Payroll Checklist work areas.
2. On the Enter Parameters page, enter a name to identify the flow.
3. In the Interface Type field, select **Oracle Fusion Time and Labor**.
4. Optionally, specify a process configuration group.

 **Tip:** If you don't need to connect, schedule, or review the flow parameters, you can skip these pages and submit the flow by clicking the **Submit** button.

5. Optionally, complete the parameters on the Flow Interaction page.
6. Optionally, complete the parameters on the Scheduling page.
7. Click the **Submit** button on the Review page. In the Confirmation window, click the **Ok and View Checklist** button.
8. On the Task Details tab of the payroll flow, select the row for the Upload File task. Select **Mark as Complete** from the Actions menu.

 **Note:** You don't need to upload a file for Time and Labor. The Upload task is only used to upload an XML file from a third-party time provider.

9. The Load Time Card Batches task starts automatically after you mark the Upload File task completed.

The View Person Process Results page lists the time card IDs included in the batch of transferred time cards. View the time entries for each person's time included in the batch using the Manage Calculation Cards task in the Payroll Calculation work area.

Importing Time Entries from a Third-Party Provider

Complete the following steps to transfer your time card entries to calculation cards:

1. Create an XML file using the file format for transferring a batch of time cards.
2. If your file is over 2 MB, stage the file on the Oracle WebCenter Content Server and note the Content Id.
3. Use the Submit a Payroll Flow task to submit the Load Time Card Batches flow from the Payroll Administration or Payroll Checklist work areas.
4. On the Enter Parameters page, enter a name to identify the flow.
5. In the Interface Type field, select **Import Time XML**.
6. If your file is over 2 MB, specify the number for the **Content ID** submission parameter.
7. Optionally, specify a process configuration group.
8. Optionally, complete the parameters on the Flow Interaction page.
9. Optionally, complete the parameters on the Scheduling page.
10. Click the **Submit** button on the Review page. In the Confirmation window, click the **Ok and View Checklist** button.
11. Click the **Submit** button on the Review page. In the Confirmation window, click the **Ok and View Checklist** button.
12. If your file is over 2 MB, skip the Upload File task, and the load batches tasks starts automatically.
13. If your file is under 2 MB, do the following:
 - a. On the Task Details tab of the payroll flow, click the **Go To Task** button for the Upload File task.
 - b. On the Upload File page in the File field, click the **Add** button to display the Manage Attachments window.
 - c. In the Type field, select **File**.
 - d. Search for and select your XML file. Click the **OK** button.

The load batches task starts automatically after the Upload File task completes.

The View Person Process Results page lists the time card IDs included in the batch of transferred time cards. View the time entries for each person's time included in the batch using the Manage Calculation Cards task in the Payroll Calculation work area. Entries transferred from third-party providers only display on the calculation cards, and not in the Oracle Fusion Time and Labor time cards.

Related Topics

- [Processing Time Entries in Payroll: Explained](#)
- [Connecting Flows: Points to Consider](#)

File Format for Importing Time Entries

You import time entries from a third-party provider by submitting the Load Time Card Batches process from the Payroll Checklist or Payroll Administration work areas. When you submit the process, you specify the batch XML file that includes your time entries. This topic explains the XML file format and XML tags you must use in the file.

You submit the Load Time Card Batches process from the Payroll Administration work area. The process creates a new calculation card or updates an existing card for each worker with time entries included in the batch.

XML File Format for Importing Time Entries

When you create a file to transfer time card entries to payroll, use the following structure.

```
<TIME_CARD_LIST>
<TIME_CARD>...
<ACTION>
<TIME_CARD_ID>
<MAPPING_ID>
<MAPPING_NAME>
<LDG_ID>
<LDG_NAME>
<HR_TERM_ID>
<TERM_NUMBER>
<HR_ASSIGNMENT_ID>
<ASSIGNMENT_NUMBER>
<LEGAL_EMPLOYER_ID>
<LEGAL_EMPLOYER_NAME>
<TIME_CARD_START>
<TIME_CARD_END>
<TIME_ITEM_LIST>...
<TIME_ITEM>
<TIME_TYPE>
{
<PAYMENT_RATE_ID>
<PAYMENT_RATE_NAME> |
<RATE_AMOUNT>
<PERIODICITY>
<FACTOR> |
<AMOUNT>
<PERIODICITY>
}
<TIME_UNIT>
<TIME_UOM>
<TIME_ITEM_START>
<TIME_ITEM_END>
<COST_SEGMENTS>
<SEGMENT1..30>
</COST_SEGMENTS>
<PROPERTIES_LIST>..
<PROPERTY_ITEM>
<NAME>
<VALUE>
</PROPERTY_ITEM>
</PROPERTIES_LIST>
</TIME_ITEM>
</TIME_ITEM_LIST>
</TIME_CARD>
```

</TIME_CARD_LIST>

XML Tags

This table describes the purpose of the tags used in the XML file.


XML Tag	Description
TIME_CARD_LIST	Parent tag that contains a set of time cards.
TIME_CARD	Object that contains the information about a specific time card.
ACTION	Action to perform, such as CREATE, REMOVE, MODIFY.
TIME_CARD_ID	Unique identifier for this time card.
MAPPING_ID	Identifier for the payroll component definition. Specify the Mapping ID or the Mapping Name. If none is included, the process uses the default interface type Import Time XML and attempts to find a mapping.
MAPPING_NAME	Name used for the mapping. Specify the mapping name or the mapping ID. If none is included, the process uses the default interface type Import Time XML and attempts to find a mapping.
LDG_NAME	Name of the legislative data group (LDG) for this record. Specify the identifier or name of the LDG. All the records in the XML file must belong to the same LDG. If you don't include the LDG_ID or the LDG_NAME, the application uses the legislative data group you entered for the Load Time Card Batches process.
LDG_ID	Identifier for the LDG for this record. Specify the identifier or name of the LDG. All the records in the XML file must belong to the same LDG. If you don't include the LDG_ID or the LDG_NAME, the application uses the LDG you entered for the Load Time Card Batches process.
TERM_NUMBER	Number that identifies the employment terms for the time entry.
ASSIGNMENT_ NUMBER	Number that identifies the employment assignment for the time entry.
TIME_CARD_START	Start date of the time card.
TIME_CARD_END	End date of the time card.
TIME_ITEM_LIST	Tag that contains a set of time items.
TIME_ITEM	Object that contains information about a specific hour item.

XML Tag	Description
TIME_TYPE	Name supplied by the time application that maps to the payroll element and calculation component.
PAYMENT_RATE_ID	Identifier for the rate definition used to calculate the payment amount.
PAYMENT_RATE_NAME	Name of the rate definition used to calculate the payment amount.
RATE_AMOUNT	Actual rate used to calculate the payroll amount.
AMOUNT	Flat amount used to calculate the rate based on periodicity.
PERIODICITY	Frequency that determines the rate value, used with amount or rate amount.
FACTOR	Multiplier applied to the derived rate to calculate the payment amount.
TIME_UNIT	Number of units for the Unit of Measure specified in TIME_UOM. For example, if the UOM is hours, 8 units is 8 hours worked.
TIME_UOM	Unit of measure for specifying time unit, such as hours.
TIME_ITEM_START	Start time for the time entry.
TIME_ITEM_END	Ending time for the time entry.
COST_SEGMENTS	List of the costing segments.
PROPERTY_LIST	Set of properties for the time item.
PROPERTY_ITEM	Additional information that is captured. For example, a value definition for the property item State would return State and the name of the State.
NAME	Name of a property for the time item.
VALUE	Value of a property for the time item.

Integrating Absence Management with Global Payroll: Procedure

Use Oracle Fusion Absence Management to set up and enroll persons in absence plans. Transfer absence information to Oracle Fusion Global Payroll for processing and paying absences.

Integrating Absence Management with payroll requires that you complete steps in different work areas for Absence Management, Human Resources, and payroll. The following table displays the sequence of tasks to perform in each work area.

Steps	Absence Administration work area	Person Management work area	Set Up and Maintenance work area
Create an absence plan, such as an Accrual, Qualification, or No Entitlement plan.	Manage Absence Plans task		
Create absence types based on predefined absence patterns and associate them to the absence plans.	Manage Absence Types task		
Create rate definitions to use in calculating accrual and liability balances.			Manage Rate Definitions task
Create absence elements.			Manage Elements task
<p>Complete the following payroll integration information:</p> <ul style="list-style-type: none"> Select the Transfer absence payment information for payroll processing check box. Select the element for the plan in the Element field. The element links the absence plan to the calculation component shown on the person's calculation card Optionally, specify rates. These rates override rates specified elsewhere. 	<p>Manage Absence Plans task</p> <p> Note: Find these settings in the Payroll Integration section of the Entries and Balances tab on the Create Absence Plan page.</p>		
<p>Enroll persons in the absence plan:</p> <ul style="list-style-type: none"> Add the plan and provide the entitlement date information. For accrual plans, select the Run Accruals for Selected Plan task action. 		Manage Absence Records task	

After setting up the information, you can record an absence using the Manage Absence Records task in the Person Management work area. When the absence is approved, the daily and summary breakdown information is automatically transferred to payroll.

You can then process the payroll that includes these absence entries and view the resulting absence balances on the person's Statement of Earnings. After you process and archive payments, you can view the resulting absence balances on the person's payslip.

Related Topics

- [How Many Salary Bases to Create: Points to Consider](#)
- [Rates Used to Calculate Absences in Payroll: Explained](#)

Creating Payroll Elements for Processing Absences: Procedure

You create elements to calculate and process absence payments in Oracle. When you create an absence element, your responses to the element template questions determine which elements, balances, formulas, and calculation components the template generates.

Defining an absence element involves the following steps:

- Creating an absence element
- Completing absence detail questions
- Completing accrual liability and balance payment questions
- Completing absence payment questions
- Submitting the element
- Creating element eligibility records and cost distributions

Creating an Absence Element

Create an absence element, selecting a primary classification of Absence, and a secondary classification. Typically, the predefined values include vacation, maternity, sickness, and other. Use the Manage Elements task in the Payroll Calculation or Setup and Maintenance work areas.

Completing Absence Detail Questions

The questions you complete in the Absence Details section determine which subsequent questions the template displays. You enter the following information in the Absence Details section:

1. Select the absence information to transfer to payroll based on the type of absence management plan.

Absence Management Plan Type	Absence Information to Transfer
Accrual	Accrual Balances
Accrual	Accrual Balances and Absences
Qualification	Qualification Absences
No Entitlement	No Entitlement Absences

Absence Management Plan Type	Absence Information to Transfer
------------------------------	---------------------------------

2. Specify the calculation units to use when reporting the absence, for example that is shown on the payslip, and statement of earnings. Typically, you select units for your reports that correspond to the units for your absence plan. If you select different units, the application uses 8 hours to convert days to hours.

Completing Accrual Liability and Balance Payment Questions

If you transfer accrual balances, complete the questions shown in the following table.

Question	Steps
Calculate absence liability?	<ol style="list-style-type: none"> 1. Select Yes, if you calculate liability. 2. Select a liability rate. Usually the rate is the same as the absence payment rate. You might select a different rate when estimating liability for billing purposes.
Does this plan enable balance payments when enrollment ends?	<ol style="list-style-type: none"> 1. Select Yes to create a final disbursement element and to maintain balances for the disbursement hours and payments. 2. Optionally, select a rate to use for the calculation.
Does this plan enable partial payment of balance?	<ol style="list-style-type: none"> 1. Select Yes to create a discretionary disbursement element and to maintain balances for disbursement hours and payments. 2. Optionally, select a rate to use for the calculation.

Completing Absence Payment Questions

Complete the following questions:

1. Select a method to reduce regular earnings if employees don't complete a time card, or the time card entries aren't used as a basis for calculating pay:
 - o Reduce regular earnings by absence payment (entitlement payment balance)
 - o Select rate to determine absence deduction amount (entitlement deduction balance)

You might select this method when the amount to deduct isn't 100 percent of the person's regular salary.

If you have standard earnings and absence elements in the same payroll run that reduce regular earnings, the payroll calculation reduces earnings in this sequence:

- a. Using absence element entries
- b. Using any standard earnings elements that reduce regular earnings

The salary balance isn't reduced beyond zero.

2. Optionally, select a rate to calculate the absence payment.

Submitting the Element

When you submit the element, the template automatically creates a base and retroactive pay element, balances, formulas, and calculation components. For some countries the template creates related elements, such as results and calculator elements.

The template also creates additional elements, depending on the options selected in the template to transfer absence information, as shown in the following table.

Type of Absence Information to Transfer	Optional Balance Payments Selected	Additional Elements Created
Accrual Balances	<ul style="list-style-type: none">Discretionary DisbursementsFinal Disbursements	<ul style="list-style-type: none">AccrualDiscretionary DisbursementFinal Disbursement
Accrual Balances and Absences	<ul style="list-style-type: none">Discretionary DisbursementsFinal Disbursements	<ul style="list-style-type: none">AccrualEntitlementDiscretionary DisbursementFinal Disbursement
Qualification Absences	none	Entitlement
No Entitlement Absences	none	Entitlement

Creating Element Eligibility Records and Cost Distributions

Create element eligibility records for all the elements generated by the template, for example for your accrual, entitlement, discretionary and final disbursement elements.

If your enterprise calculates cost distributions, specify costing for all the element eligibility records. For example, for an accrual element, you do the following steps

1. Create element eligibility records for the accrual, accrual results, accrual retroactive, and accrual retroactive results elements.
2. Specify costing for the accrual results and retroactive results elements.

The costing process would cost the change in the liability balance since the last payroll period, debit the expense account and credit the liability account.

Related Topics

- Costing of Elements: Critical Choices
- Creating Payroll Elements for an Absence Accrual Plan: Worked Example
- Rates Used to Calculate Absences in Payroll: Explained

Importing Absence Entries to Payroll: Procedure

You can import worker absences from Oracle Fusion Absence Management or another absence application to Oracle Fusion Global Payroll to process absence payments. When you set up an absence plan, you associate an absence element to transfer the absence information to a person's calculation card. Processing absence information for payroll includes the following aspects.

- Importing absence entries
- Processing absence entries
- Validating absence results
- Correcting absence results

Importing Absence Entries from Absence Management

Recording an absence in Absence Management transfers a summary record and a daily breakdown for each day the person is absent to the person's calculation card. The summary record contains information such as the start and end date of the absence, the absence rate, the units. The daily breakdown shows the absence date and any factor to be applied, such as a percentage to use when calculating the absence payment.

Importing Absence Entries from a Third-Party

If you use a third-party absence application, you create an XML file for the absence entries and transfer it to payroll using the Load Absence Batches process. Complete the following steps:

1. Extract your absence data into an XML file. You must use the required file format and XML tags.
2. Use the Submit a Payroll Flow task to submit the Load Absence Batches flow from the Payroll Administration or Payroll Checklist work areas.
3. On the Enter Parameters page, enter a name to identify the flow.
4. In the Interface field, select **Import Absence XML**.
5. Optionally, specify a process configuration group.
6. Optionally, complete the parameters on the Flow Interaction page.
7. Optionally, complete the parameters on the Scheduling page.
8. Click the **Submit** button on the Review page. In the Confirmation window, click the **Ok and View Checklist** button.
9. On the Task Details tab of the payroll flow, click the **Go To Task** button for the Upload File task.
10. On the Upload File page in the File field, click the **Add** button to display the Manage Attachments window.
11. In the Type field, select **File**.
12. Search for and select your XML file. Click the **OK** button.

The load batches task starts automatically after the Upload File task completes.

The View Person Process Results page lists the absence IDs included in the batch of transferred absences. View the absence entries for each person included in the batch using the Manage Calculation Cards task in the Payroll Calculation work area.

Processing Absence Entries

Transferring the absence information to a calculation card creates an element entry for the element associated to the absence plan. The payroll run processes this entry using the formula attached to the element and values from the calculation card. The resulting values are then passed back to the element entry and stored as run results and balances.

Validating Absence Results

You can verify the results of absences after calculating run results by using payroll reports or the statement of earnings. After archiving payment results, validate absence information on the Payroll Register Report or the person's payslip.


Review absence results on the following payroll reports:

- Element Result Report
- Gross-to-Net Report
- Payroll Activity Report
- Payroll Balance Report
- Payroll Register Report

Use the View Person Process Results task in the Payroll Calculation work area or click the person's record on the Person Process Results page of the payroll calculation flow to go to the person's statement of earnings. The following table lists the absence information displayed on the statement of earnings.

Statement of Earnings Sections	Information Displayed
Absences Accrual tab in the Quick Reference Summary	Accrual balances for accrual days and accrual hours
Absences	<ul style="list-style-type: none"> • Total for the entitlement, discretionary disbursement, final disbursement payment • Absence deductions made for that period

Use the View Payslip task to view the person's absence information in the Person Management work area. The following table lists the absence information displayed on the payslip.

Payslip Sections	Information Displayed
Summary	Total amount for the following balances: entitlement payment, entitlement deduction, final disbursement payment, and discretionary disbursement payment
Absence	Payment balances that contribute to the total absence balance shown in the Summary region
	 Note: Subregions display hours and rate, and days and rate details for these balances
Absence Accrual	Balances for accrual days and accrual hours

For accrual liabilities, the Employer Charges section of the Statement of Earnings page and payslip display the difference for liability balance since the last payroll.

Correcting Absence Results

You can continue to transfer updated absence entries to a person's calculation card until the payroll calculation starts. Any further updates are then processed as retroactive absences entries in the following payroll run.

File Format for Importing Absence Entries

When you submit the Load Absence Batches process, you specify the attachment for the XML file that contains the absence data. This topic explains the XML file format and XML tags you must use in the file.

You submit the Load Absence Batches process from the Payroll Administration work area. The process creates a new calculation card or updates an existing card for each worker whose absence information is transferred.

XML File Format for Importing Absence Information to Payroll

When you create a file to transfer absence information to payroll, use the following format.

```
<ABSENCE_LIST>
  <ABSENCE>
    <ABSENCE_TYPE>
    <ACTION>
    <ABSENCE_ID>
    <MAPPING_ID>
    <MAPPING_NAME>
    <LDG_ID>
    <LDG_NAME>
    <HR_TERM_ID>
    <TERM_NUMBER>
    <HR_ASSIGNMENT_ID>
    <ASSIGNMENT_NUMBER>
    <ABSENCE_RATE_ID>
    <ABSENCE_RATE_NAME>
    <ABSENCE_UNIT>
    <ABSENCE_UOM>
    <ADJUSTMENT_UNIT>
    <FACTOR>
    <CALCULATION_DATE>
    <PERIODICITY>
    <ABSENCE_START>
    <ABSENCE_END>
    <ABSENCE_DATE_LIST>
    <ABSENCE_DATE>
    <LEAVE_DATE>
    <ACCRUED_DATE>
    <OVERRIDING_FACTOR>
    <OVERRIDING_RATE_ID>
    <OVERRIDING_RATE_NAME>
    <OVERRIDING_UOM>
    <OVERRIDING_UNIT>
  <\ABSENCE_DATE>
  <\ABSENCE_DATE_LIST>
  <\ABSENCE>
<\ABSENCE_LIST>
```

XML Tags

This table describes the purpose of the tags used in the XML file.

XML Tag	Description
ABSENCE_LIST	Outermost tag that contains a set of absences.
ABSENCE	Tag containing information about a particular absence.
ABSENCE_TYPE	Type of absence that is being transferred to payroll, such as accrual, accrual with entitlement, or entitlement.
ACTION	Type of action that will be performed, such as CREATE, REMOVE, and MODIFY.
ABSENCE_ID	Unique identifier for the absence from the source application. Never use the same ID twice to identify another absence.
MAPPING_ID	Identifier for the payroll component definition, which is used to create the absence in payroll.
MAPPING_NAME	Name used for the mapping.
LDG_ID	ID of the legislative data group associated with the record.
LDG_NAME	Name of the legislative data group associated with the record.
HR_TERM_ID	Unique ID for the HR Terms. You can provide either the TERM_NUMBER or the HR_ TERM_ID. If you provide the TERM_NUMBER then you must also provide the legal employer details.
TERM_NUMBER	Number that identifies the employment terms for the absence.
HR_ASSIGNMENT_ID	This is the HR Assignments unique ID. You can provide either the ASSIGNMENT_ NUMBER or the HR_ ASSIGNMENT_ID. If you provide the ASSIGNMENT_ NUMBER then you must also provide the legal employer details.
ASSIGNMENT_ NUMBER	Number that identifies the employment assignment for the absence.
ABSENCE_ RATE_ID	Unique identifier for the absence rate.
ABSENCE_ RATE_NAME	Name of the rate used to calculate the payment amount.
ABSENCE_UNIT	Unit of time in which the absence is recorded.
ABSENCE_UOM	Unit of measure being used for the absence (for example, days, hours or weeks).
ADJUSTMENT_ UNIT	Unit of time in which an adjustment is being made to the absence.
FACTOR	Factor that is used in the calculation of the absence.

XML Tag	Description
CALCULATION DATE	Date used for payroll calculations, such as the payment calculation for maternity leave based on the baby's due date.
PERIODICITY	Used with the amount or rate, the periodicity is the frequency that determines the absence rate.
ABSENCE_START	Date the absence started.
ABSENCE_END	Date the absence ended.
ABSENCE_ DATE_LIST	List of dates in which the absence occurred.
ABSENCE_DATE	Date the absence is being reported.
LEAVE_DATE	Date on which the leave of absence occurred.
ACCRUED_DATE	Date on which the absence was accrued.
OVERRIDING_ FACTOR	Factor that is being used to override the calculation of the absence.
OVERRIDING_ RATE_ID	Unique identifier for the rate being used to override the absence.
OVERRIDING_ RATE_NAME	Name of the overriding rate that will be used to calculate the absence.
OVERRIDING_UOM	Unit of measure being used to override the absence (for example, days, hours or weeks).
OVERRIDING_ UNIT	Unit of time in which an override is being made to the absence.

Running the Load Benefit Batches Process: Procedure

Use the Load Benefit Batches process to transfer pension deduction information from a benefits application to benefits and pensions calculation cards for payroll processing.

Submitting the Process

You can submit the process from the Payroll Checklist or Payroll Administration work areas, or you can add it to a payroll flow pattern so that it runs as part of your regular payroll flow.

To submit the process:

1. Extract your pension data into an XML file, which must use the required file format and XML tags.
2. Use the Submit a Payroll Flow task to submit the Load Benefit Batches flow from the Payroll Administration or Payroll Checklist work areas.
3. On the Enter Parameters page, enter a name to identify the flow.
4. In the Interface field, select **Import Benefit XML**.

5. Optionally, specify a process configuration group.
 6. Optionally, complete the parameters on the Flow Interaction page.
 7. Optionally, complete the parameters on the Scheduling page.
 8. Click the **Submit** button on the Review page. In the Confirmation window, click the **Ok and View Checklist** button.
 9. On the Task Details tab of the payroll flow, click the **Go To Task** button for the Upload File task.
 10. On the Upload File page in the File field, click the **Add** button to display the Manage Attachments window.
 11. In the Type field, select **File**.
 12. Search for and select your XML file. Click the **OK** button.
- The load batches task starts automatically after the Upload File task completes.

Resolving Transfer Errors

When you load a benefits batch, the application validates the entries to confirm that the worker isn't terminated and is eligible for the deduction. The application rejects entries for any date beyond the worker's termination date.

If the process ends in error, you can roll it back, resolve the error in the source application, and resubmit the Load Benefit Batches process.

Viewing and Correcting Entries

When the process completes successfully, it creates or updates a calculation card for each person included in the XML file. Use the Manage Calculation Cards task in the Payroll Calculation work area to view the new entries on the cards.

You can enter or update the following values on the cards, if required:

- Payee
- Reference Number
- Employee Additional Contribution

The other values are view-only and must be maintained in the source application.

Related Topics

- [Entering Calculation Values for Pensions: Points to Consider](#)

File Format for Importing Pension Deductions

When you submit the Load Benefit Batches process, you specify the attachment for the XML file that contains the benefit data. This topic explains the XML file format and XML tags you must use in the file. You submit the Load Benefit Batches process from the Payroll Checklist or Payroll Administration work areas. The process creates a new calculation card or updates an existing card for each worker whose pension information is transferred.

XML File Format for Importing Pension Deductions to Payroll

When you create a file to transfer pension deduction information to payroll, use the following format.

```
<BENEFIT LIST>
<BENEFIT>...
<ACTION>
```

```
<BENEFIT_ID>
<MAPPING_ID>
<LDG_ID>
<LDG_NAME>
{
  <HR_TERM_ID>
  <TERM_NUMBER>
  <HR_ASSIGNMENT_ID>
  <ASSIGNMENT_NUMBER>
}
<LEGAL_EMPLOYER_ID>
<LEGAL_EMPLOYER_NAME>
<BENEFIT_START>
<BENEFIT_END>
{
  <BENEFIT_RATE_ID>
  <BENEFIT_RATE_NAME> |
  <AMOUNT>
  <PERIODICITY>
  <BENEFIT_MAX_ELECTION>
  <BENEFIT_REF_NUMBER>
}
</BENEFIT>
</BENEFIT_LIST>
```

XML Tags

This table describes the purpose of the tags used in the XML file.

XML Tag	Description
BENEFIT_LIST	Outermost tag that contains a set of benefits.
BENEFIT	Tag containing information about a particular benefit.
ACTION	The type of action that will be performed, such as CREATE, REMOVE, and MODIFY.
BENEFIT_ID	Unique identifier for the benefit from the source application. Never use the same ID twice to identify another benefit.
MAPPING_ID	Identifier for the payroll component definition, which is used to create the benefit in payroll.
LDG_ID	ID of the legislative data group associated with the record.
LDG_NAME	Name of the legislative data group associated with the record.
HR_TERM_ID	Unique ID for the employment terms. You can provide either the TERM_NUMBER or the HR_TERM_ID. If you provide the TERM_NUMBER then you must also provide the legal employer details.
TERM_NUMBER	Number that identifies the employment terms for the pension deduction.

XML Tag	Description
HR_ASSIGNMENT_ID	Unique ID for the assignment. You can provide either the ASSIGNMENT_ NUMBER or the HR_ASSIGNMENT_ID. If you provide the ASSIGNMENT_ NUMBER then you must also provide the legal employer details.
ASSIGNMENT_ NUMBER	Number that identifies the employment assignment for the pension deduction.
LEGAL_ EMPLOYER_ID	ID of the legal employer name that the term or assignment belongs to.
LEGAL_ EMPLOYER_NAME	Legal employer name that the term or assignment belongs to.
BENEFIT_START	Start date of the benefit.
BENEFIT_END	End date of the benefit.
BENEFIT_ RATE_ID	ID of the rate that will be used to calculate the payment amount.
BENEFIT_ RATE_NAME	Name of the rate that will be used to calculate the payment amount.
AMOUNT	Amount that is used to calculate the rate using the periodicity.
PERIODICITY	Used with the amount or rate, the periodicity is the frequency that determines the rate value.
BENEFIT_ MAX_ELECTION	Annual maximum election amount that can be processed.
BENEFIT_ REF_NUMBER	Employee's reference number with the provider of the pension (benefit organization).

25 HCM Extracts

Managing Extracts

Extract Components: How They Work Together

The HCM Extracts feature is a flexible tool for generating data files and reports. This topic covers how you can use the extract components to define what information you want the application to extract and report on. It also explains how the application displays, formats, and delivers the information.

Extract Definitions

An extract definition refers to the complete setup of an extract, that consists of extract data groups, criteria, records, attributes, advanced conditions and output delivery options. An extract definition consists of:

- One or more extract data groups, depending on how many logical entities you want to extract.
- One or more extract records depending on how many groups of information you want to collect.
- One or more attributes depending on how many individual fields of data you want to collect.

You use HCM extracts to extract, archive, transform, report, and deliver high volumes of HCM data from the Fusion HCM database. You can generate the output in the following formats:

- CSV
- XML
- Excel
- HTML
- RTF
- PDF

You can distribute the extracted information by e-mail, fax and other delivery modes. Some common examples of extracts are: PDF payslips delivered to employees' mailboxes, payroll or benefits data transferred to third-party service providers, HR and talent data exchange between Fusion and legacy applications, for example in a coexistence scenario.

Data Groups

Extract data groups represent a business area or logical entity, for example person, assignment, or benefits. The application uses this information to retrieve the database item groups. You define one data group as the primary or root data group and this data group is the starting point of the data extraction.

Extract data group connections capture the association details between the current data group and the parent data group. The data group connections form the hierarchical relationship among the data groups.

You can define a set of filtering conditions the application must perform on an extract data group using the extract data group criteria. You specify the criteria conditions using an expression or fast formula.

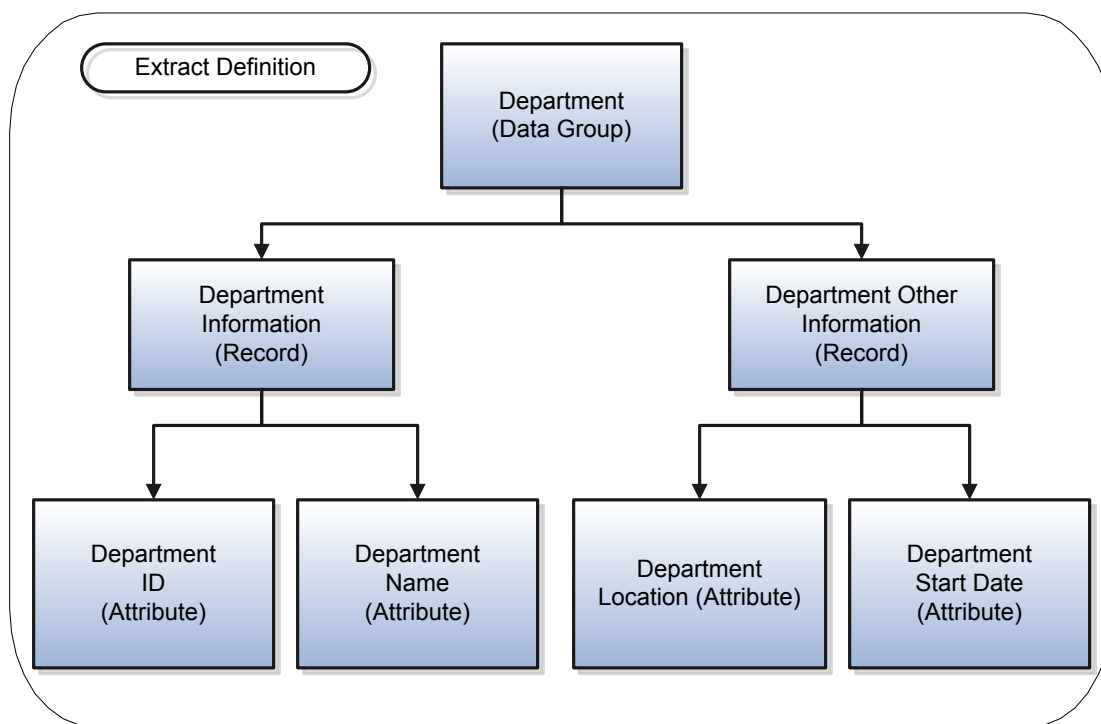
Extract Records

Extract records represent a grouping of related data or a physical collection of all fields required in the extract. For example, the Employee data group can have records such as Basic Details, Pay Details, Location Details, and Primary Contact. An extract record is a collection of attributes which you can organize in a required sequence. For example, if a data group has 3 records, then you can specify the sequence in which the application processes the records. You can also select the next data group to identify which data group the application processes next.

Attributes

Attributes are the individual fields inside the extract record. An attribute is the lowest attribute level of a HCM extract and represents a piece of information, for example, person first name, person last name or person date of birth.

This figure demonstrates the hierarchy of information within a data group definition.



Related Topics

- [Generating Flexfield Database Items: Explained](#)

HCM Extract Types: Explained

The type of extract you select determines the purpose of the extract. It also determines the parameters that are automatically generated. For example, if you select the Payroll Interface extract type, then the application creates a changes only parameter, as well as the other parameters. You can select the extract type on the Manage HCM Extract Definitions page.

This table lists the different extract types and why you select them.

Extract Type	Purpose
Full Profile	Use for complete employee and payroll data archives.
Payroll Interface	Use for providing data to third party payroll service providers.
Payments	Use for salary payment method archives. For example, Paid through cheque or bank transfer.
Benefit Carrier	Use for providing data to third party benefits service providers.
Archive Retrieval	Use for reports based on permanently archived data, for example, payslip.
EOY Archive	Use for end of year archives (HR, Benefits).
HR Archive	Use for all HR archives.
Payroll Archive	Use for all payroll or payslip archives.
Other Payroll Archive	Use for all payroll archives.

Payroll Interface Extract Definitions: Overview

The extract definitions of the Payroll Interface extract type determine the data you send your third-party payroll provider. The following table lists the predefined Payroll Interface extract definitions that you can use or customize to meet your business-specific extract requirements.

Extract Definition	Purpose	Output
Global Payroll Interface	Derives payroll data either from element entry values or from balance results created by the Calculate Gross Earnings process.	eText and XML
Payroll Interface for NGA	<p>Extracts general-purpose HR and payroll-related data to integrate with Payroll Exchange for third-party payroll processing by NGA Human Resources. This extract transforms the data into an HR-XML format compliant with NGA Human Resources standards.</p> <p>Use the predefined Run Payroll Interface Report for NGA process if you don't require special customizations.</p>	XML
US ADP PayForce Third-Party Ad-Hoc Extract	Derives payroll data from element entries, including common HR and payroll data for a date range or a payroll period. Output format is compliant with Automatic Data Processing (ADP) PayForce standards.	eText and XML

Extract Definition	Purpose	Output
US ADP PayForce Third-Party Periodic Extract	Derives payroll data from balance results created by the Calculate Gross Earnings process for a payroll period. Output format is compliant with ADP PayForce standards.	eText and XML

Related Topics

- [Choosing Predefined Extract Definitions for Payroll Interface: Critical Choices](#)
- [Creating Extract Definitions for Payroll Interface: Procedure](#)
- [Extracting Payroll-Related Data: Critical Choices](#)
- [Extracting Payroll Data for Third-Party Processing: Worked Example](#)

Viewing User Entity Details: Explained

When creating an extract definition, you need user entity details to create data groups. You can access the View User Entity Details UI using the Data Exchange tasks pane or when you create a data group in the Design tab of the Manage Extract Definitions task.

Using the View User Entity Details page, you can:

- View the database items available in a user entity.
- Validate user entities for use with root data groups.
- Review the type of user entity, for example, single row, multiple rows, historic, or effective dated.
- Review the SQL query the user entity uses.
- Calculate the number of rows in a data set.
- Review the contexts used and set by the user entity.
- Verify the extracts that are using the user entity.

The following sections detail the information that you can derive from the View User Entity Details UI.

Reviewing the User Entity Details

Search for a user entity in the View User Entity Details UI and click the User Entity Details tab.

If the value of Valid for Root Data Group is Yes, the user entity can be used as a root data group. You can also determine if a user entity is valid for use as a root data group, if the value of Multiple Rows is Yes and Context Required is No. For example, the PER_EXT_SEC_PERSON_UE user entity can be used as a root data group, as the value of Multiple Rows is Yes and Context Required is No.

If a user entity is not a multiple row user entity, you must not use it as the root data group. If a multiple row user entity sets the contexts required for a single row user entity, the database item groups from the single row user entity are generally included as derived database item groups to the multiple row user entity. For example, the PER_PER_PERSON_DETAILS_UE user entity is not a multiple row user entity.

If a user entity is historic, it retrieves data irrespective of the effective date. For example, the PER_EXT_ASSIGNMENT_BASIC_HISTORY_UE user entity retrieves the entire historic details of an assignment.

If a user entity is not historic, it retrieves data as of the effective date. For example, the PER_EXT_PAY_EMPLOYEES_UE user entity retrieves Person, Assignment, and Payroll details as of the effective date.

Reviewing the SQL query

In the View User Entity Details UI, search for a user entity and click the Query tab. This tab lists the SQL query that the user entity uses to extract data. You can review the table structure and aliases which is useful for creating advanced filter criteria. If the user entity is valid as a root data group, you can calculate the rows that the SQL query will return. However, the number of rows returned is an estimate and may not match the exact number of rows the extract will return. The actual extract output depends on the filter criteria, expressions, and fast formulas used in the extract.

Reviewing the User Entity Contexts

In the View User Entity Details UI, search for a user entity and click the Context tab. The Contexts Required section lists the contexts that you have to set to use the user entity in the extract. For example, the PER_PER_PHONES_UE user entity requires the PERSON_ID context; therefore, you must set the PERSON_ID context to use this user entity. Generally, the user entities that are not historic require the EFFECTIVE_DATE context. However, the EFFECTIVE_DATE context is set by default, so you do not have to set this context explicitly. The Contexts Set section lists the contexts that are set by the user entity. For example, the PER_EXT_SEC_PERSON_UE user entity sets the PERSON_ID, ORGANIZATION_ID, and ENTERPRISE_ID contexts that can be used by other user entities.

Verifying Extracts Using the User Entity

In the View User Entity Details UI, search for a user entity and click the Extracts Using User Entity tab. If another extract is using the user entity, you can review the extract design to understand how the user entity is being used. You can also copy the extract design and then modify it to suit your needs.

Frequently Used User Entities

A user entity is a logical entity associated with a data group defined using HCM extracts. This topic describes the frequently used user entities and the type of data you can extract by using those user entities. You select user entities in the application when you define a data group.

The following table lists the most frequently used user entities.

User Entity Name and Code	Description
Person (PER_EXT_SEC_PERSON_UE)	Retrieves all persons across the enterprise and all person related attributes.
Worker Payroll (PER_EXT_PAY_EMPLOYEES_UE)	Retrieves all workers and their payrolls across the enterprise, all person, worker, payroll related attributes, and element entry data.
Extract Assignment Basic History (PER_EXT_ASSIGNMENT_BASIC_HISTORY_UE)	Retrieves assignment history as on the assignment effective start date.
Extract Assignment Basic Information (PER_EXT_SEC_ASSIGNMENT_BASIC_UE)	Retrieves assignment data as on the effective date.
Assignments Range (PER_EXT_SEC_ASSIGNMENT_RANGE_UE)	Retrieves assignment history as on the effective date.

User Entity Name and Code	Description
Extract Current and Future Persons (PER_EXT_SEC_PERSON_NOW_FUTURE_UE)	Retrieves current and future person details. Specify advanced filter criteria to restrict person types.
Extract Current and Future Assignments (PER_EXT_SEC_ASSIGNMENT_NOW_FUTURE_UE)	Retrieves current and future assignments.

You can view more details about the user entities using the View User Entity Details task.

Related Topics

- [Generating Flexfield Database Items: Worked Example](#)

HCM Extracts Templates: Explained

Use the predefined Work Structure and Worker extract templates to quickly set up extracts that generate work structure and worker details. The Work Structure extract generates details of business units, legal entities, departments, grades, jobs, locations, and positions. The Worker extract generates details of person names, phones, e-mails, national identifiers, address, legislative data, accruals, absences, work relation, work term, assignment, work measure, supervisor, and salary.

To use these extracts, copy and configure them to suit your needs. You must not run the extract template directly. These extracts are complex and detailed. They can extract a huge volume of data and can be time consuming as they are designed on historic user entities to return all the details irrespective of the effective date.

When using these extract templates, ensure the following to avoid unneeded and time consuming data retrieval:

- Define filter criteria. For example, if you want to extract data related to specific locations, include an expression or fast formula in the data group filter conditions.
- Remove unwanted data groups. For example, if you are not interested in positions, then you can remove the positions data group.
- Review all the details to ensure all the information is relevant to your needs.

Using Work Structure and Worker Extract Templates

To use the Worker or Work Structure extract templates:

1. Copy the Worker or Work Structure extract definition to create a custom extract.
2. Update the copied extract.
3. Generate and compile fast formulas.
4. Validate the extract design.
5. Refine the parameters.

Defining an Extract: Worked Example

This example demonstrates the steps required to create an extract definition using the Desktop interface. You can access the Desktop interface by clicking the Switch Layout button on the Manage HCM Extract Definitions page. Before you create an extract definition, you should understand the following details:

- Information that you want to extract
- Structure in which the data must be extracted
- How you want to deliver this data (including file format, delivery mechanism, and frequency information)

FAST bank is a global organization with subsidiaries all over the world. As part of an external business reporting requirement, FAST bank is required to extract the department and employee details (grouped by department) across the entire company. This information must be sent to a third party in an XML file and to HR Managers in a PDF file using e-mail. The following table summarizes the key decisions in this scenario:

Decisions to Consider	In This Example
How many extracts do I create to produce this type of report?	You create one extract definition to define a headcount report.
How many data groups do I create?	There are 2 functional groups of information, therefore you create two data groups, one for department and one for employees.
How many records do I create?	<p>You decide the number of records based on the subgroup of attributes within a data group. In this example, you create two records for the department data group:</p> <ul style="list-style-type: none"> • Department Details • Department Summary <p>You create one record for the employees data group: Employee Details.</p>
How many attributes do I create?	<p>You decide the number of attributes based on the specific information required for that report. Create the following attributes for the Department Details record:</p> <ul style="list-style-type: none"> • Department Name • Department Location <p>For the Department Summary record, create the following attributes:</p> <ul style="list-style-type: none"> • Record Code • Report Date • Employee Count <p>For the Employees Details record, create the following attributes:</p> <ul style="list-style-type: none"> • Full Name • Gender • Date of Birth • Salary • Bonus

Decisions to Consider	In This Example
	<ul style="list-style-type: none">Tax Rate
Do I create any fast formulas?	<p>You can use fast formulas at the following levels:</p> <ul style="list-style-type: none">Extract Criteria level to determine certain conditions.Extract Rule level to derive attribute values.Extract Advanced Condition level to specify complex conditions.Extract Record level to automatically generate formulas when you use the Generate Formula option.

Creating an Extract Definition

- On the Manage HCM Extract Definitions page click on the Create icon.
- Use the Switch Layout button to open the extract in the Desktop interface.
- Enter 01-Jan-2000 as the Session Effective Date.

The session effective date is an effective start date that applies to all date-effective interactions in the current session.
- Enter FAST Bank Extract as the name and select HR Archive as the type. The application automatically creates the tag name based on the extract name and uses this name to generate the XML output file.
- Click Save and the application saves the extract definition and automatically generates the parameters based on the type of extract. The parameters control the output of an extract. In this example, the application creates the following parameters:
 - Effective Date
 - Legislative Data Group
 - Parameter Group
 - Report Category
 - Request ID
 - Start Date

Creating Extract Data Groups

- Select the Extract Data Group link from the navigation tree to open the Extract Data Groups region.
- Click on Create to define a new data group. A data group represents data that belongs to one or more logical data entities.
- Complete the general fields, as shown in this table:

Field Name	Entry
Name	Departments
User Entity	PER_EXT_SEC_ORGANIZATION_UE
Root Data Group	Yes (By selecting this option you select this data group as the starting point for the extract execution.)

- Select Save and Create Another to create a data group for Employees.
- Complete the general fields, as shown in this table:

Field Name	Entry
Name	Employees
User Entity	PER_EXT_SEC_ASSIGNMENT_UE
Root Data Group	No

Creating Extract Data Group Connections

1. Select Extract Data Group in the navigation tree to display the data groups in a table.
2. Select the Employees Data Group and define the data group connection details.
3. Complete the general fields, as shown in this table:

Field Name	Entry
Parent Data Group	Departments
Parent Data Group Database Item	PER_EXT_ORG_ORGANIZATION_ID
Data Group Database Item	PER_EXT_ASG_ORG_ID

4. Define the data group criteria for each data group.

Creating Extract Records

1. Select the Departments Data Group from the navigation tree and select the Create icon in the Extract Records region. Extract records represent a physical collection of all required fields. If a data group has 3 records, then you can specify the sequence in which the application processes the records using the sequence field. You can also select the Next Data Group to identify which data group the application processes next.
2. Create two records for the Departments Data Group.
3. Complete the general fields, as shown in this table:

Field	Department Summary	Department Details
Name	Department Summary	Department Details
Effective Start Date	1/1/00	1/1/00
Sequence	20	10
Type	Trailer Record	Header Record
Process Type	Fast Formula	Fast Formula
Next Data Group	NA	Employees

4. Save the records, then select the Employees Data Group and select the Create icon in the Extract Records region.
5. Create one record for the Employees Data Group.
6. Complete the general fields, as shown in this table:

Field Name	Entry
Name	Employee Details
Effective Start Date	1/1/00
Sequence	10
Type	Detail Record
Process Type	Fast Formula

Creating Attributes

1. Select the Departments Data Group from the navigation tree and select the Department Details record.
An extract attribute is an individual field of a record.
2. Create the following extract attributes for the Department Details record and select Save.
3. Complete the general fields, as shown in this table:

Field Name	Attribute Entry	Attribute Entry
Name	Department Name	Department Location
Type	Database item group	Database item group
Database Item Group	Organization Name	Organization Location Country

4. Save the record, then select the Department Summary record.
5. Select the Create icon in the Extract Attributes region.
6. Create the following extract attributes for the Department Summary record and select Save.
7. Complete the general fields, as shown in this table:

Field Name	Attribute Entry	Attribute Entry	Attribute Entry
Name	Record Code	Report Date	Employee Count
Data Type	Text	Date	Number
Type	String	Parameter Element	Summary Element
String Value	999	NA	NA
Parameter	Effective Date	NA	NA

Field Name	Attribute Entry	Attribute Entry	Attribute Entry
Aggregate Function	NA	NA	Count
Aggregate Record Name	NA	NA	Employees Employee Details

8. Select the Employees Data Group from the navigation tree and select the Employee Details record.
9. Create the following extract attributes for the Employee Details record and select Save.
10. Complete the general fields, as shown in this table:

Field Name	Attribute Entry	Attribute Entry	Attribute Entry
Name	Full Name	Gender	Date of Birth
Start Date	1/1/00	1/1/00	1/1/00
Data Type	Text	Text	Date
Type	Database Item Group	Decoded database item group	Database item group
Database Item Group	Person Full Name	Person Gender	Person Date of Birth

Field Name	Attribute Entry	Attribute Entry	Attribute Entry
Name	Salary	Bonus	Tax rate
Start Date	1/1/00	1/1/00	1/1/00
Data Type	Number	Number	Text
Type	Database item group	Record Calculation	Rule
Database Item Group	Assignment Salary Amount	NA	NA
Calculation Expression	NA	Salary * 0.5	NA
Rule	NA	NA	FAST Bank Tax Rule

Defining the Delivery Options

1. Navigate to the Extract Execution Tree to validate the extract definition setup.
2. Select Export XSD to download the XML Schema Definition (.XSD) file for this extract setup. This exported file contains the structure of the extract definition: the data groups, records, and attributes.
3. Select the Extract Delivery Options region to define the formatting and layout options for the extract definition.
4. Complete the general fields, as shown in this table:

Field	Value	Value
Start Date	1/1/00	1/1/00
End Date	12/31/12	12/31/12
BI Publisher Template	ReportLayout	EFTLayout
Output Type	PDF	EFT
Delivery Type	E-mail	FTP
Delivery Option Name	E-mail to HR	FTP to 3rd Party
Output Name	HeadcountReport	EFTReport

5. Define further information for each delivery option in the Additional Details region. For example, add the server, username and password for the FTP delivery type.
6. Enter FAST Bank Extract as the reporting category and click Submit.

Submitting an Extract

An extract definition automatically creates an extract process (payroll flow) with the same name as the extract. The extract process enables you to define an execution sequence of multiple tasks, including before and after tasks.

1. Select the Submit Extracts task and select the FAST Bank Extract process.
2. Select Next.
3. Enter FAST Bank Extract - Jan 2012 as the Payroll Flow (extract process).
4. Enter 1/1/15 as the End Date.
5. Select Next. You can specify interaction details if the task is dependent on other tasks with different extract processes. For example, this task must wait because another task is running.
6. Select Next and review the extract. You can schedule the extract, or run it immediately.
7. Select Submit.
8. Select OK and View Checklist to view the status of the process.
9. Select the View Extract Results task to review the results of the extract run. Search for the FAST Bank Extract process.
10. Select Go to Task for FAST Bank Extract - Jan 2012, click the eyeglasses, and view the report output by selecting the report name.

Related Topics

- [How do I create a BI Publisher template for HCM extract?](#)

Using the Simplified Interface: Worked Example

This example topic demonstrates how to create a HCM extract including creating data groups, records, and attributes using the Simplified interface. FAST Bank is a global organization with subsidiaries all over the world. As part of an external reporting requirement, FAST Bank must obtain the department and employee details across the entire company. This information must

be sent to a third party in an XML file and to the HR manager with employee details grouped by department as a Headcount Report.

The following table summarizes the key decisions in this scenario:

Decisions to Consider	In This Example
How many extracts do I create to produce this type of report?	You create one extract definition to define a headcount report.
How many data groups do I create?	There are 2 functional groups of information, therefore you create two data groups, one for department and one for employees.
How many records do I create?	<p>You decide the number of records based on the subgroup of attributes within a data group. In this example, you create two records for the department data group:</p> <ul style="list-style-type: none"> • Department Details • Department Summary <p>You create one record for the employees data group: Employee Details.</p>
How many attributes do I create?	<p>You decide the number of attributes based on the specific information required for that report. Create the following attributes for the Department Details record:</p> <ul style="list-style-type: none"> • Department Name • Department Location <p>For the Department Summary record, create the following attributes:</p> <ul style="list-style-type: none"> • Record Code • Report Date • Employee Count <p>For the Employees Details record, create the following attributes:</p> <ul style="list-style-type: none"> • Full Name • Gender • Date of Birth • Salary • Bonus • Tax Rate
Do I create any fast formulas?	<p>You can use fast formulas at the following levels:</p> <ul style="list-style-type: none"> • Extract Criteria level to determine certain conditions. • Extract Rule level to derive attribute values. • Extract Advanced Condition level to specify complex conditions. • Extract Record level to automatically generate formulas when you use the Generate Formula option.

Creating an Extract Definition

1. On the Manage Extract Definitions page, click Create.

2. Enter FAST Bank Extract as the name, 01-JAN-2010 as the Start Date, and select HR Archive as the type. The application uses this name to generate the XML output file.
3. Click Save. The application saves the extract definition and automatically generates the parameters based on the type of extract. The parameters control the output of an extract.

Use the Switch Layout button to open the extract in the Desktop interface. Use the Desktop interface to create and define HCM extracts without using a drag and drop system. You can perform most of the tasks for defining the extract in the Simplified interface.

Creating Extract Data Groups and Records

1. Select the Design icon to create the data groups and records.
2. Select the Create icon or use the HCM Data Objects tree to drag and drop a data group into the local area.
3. Using the drag and drop action, drag the Organizations object into the local area. This action automatically creates an associated record object.
4. Update the data group with the following information:

Field Name	Entry
Name	Departments
User Entity	PER_EXT_SEC_ORGANIZATION_UE
Root Data Group	Yes (By selecting this option you select this data group as the starting point for the extract execution.)

5. Update the record object and create another using the following information.
6. Complete the general fields, as shown in this table:

Field	Department Summary	Department Details
Name	Department Summary	Department Details
Effective Start Date	1/1/00	1/1/00
Sequence	20	10
Type	Trailer Record	Header Record
Process Type	Fast Formula	Fast Formula
Next Data Group	NA	Employees

7. Select Save and create another data group by dragging the Employees data object to the local area. This action automatically creates an associated record object.
8. Update the data group with the following information:

Field Name	Entry
Name	Employees

Field Name	Entry
User Entity	PER_EXT_SEC_ASSIGNMENT_UE
Root Data Group	No

9. Update the record object with the following information.
10. Complete the general fields, as shown in this table:

Field Name	Entry
Name	Employee Details
Effective Start Date	1/1/00
Sequence	10
Type	Detail Record
Process Type	Fast Formula

Creating Extract Data Group Connections

1. Select the Link icon on the Employees data group, and select the Add icon to add a connection.
2. Complete the general fields, as shown in this table:

Field Name	Entry
Parent Data Group	Departments
Parent Data Group Database Item	PER_EXT_ORG_ORGANIZATION_ID
Data Group Database Item	PER_EXT_ASG_ORG_ID

3. Define the data group filter criteria for each data group by selecting the Filter icon.

Creating Attributes

1. Select the Departments Details record in the Department data group, and within the Configure icon select the Create Attribute option.
2. Complete the general fields, as shown in this table:

Field Name	Attribute Entry	Attribute Entry
Name	Department Name	Department Location
Type	Database item group	Database item group

Field Name	Attribute Entry	Attribute Entry
Database Item Group	Organization Name	Organization Location Country

3. Select the Department Summary record, and using the above method enter the following extract attribute details.
4. Complete the general fields, as shown in this table:

Field Name	Attribute Entry	Attribute Entry	Attribute Entry
Name	Record Code	Report Date	Employee Count
Data Type	Text	Date	Number
Type	String	Parameter Element	Summary Element
String Value	999	NA	NA
Parameter	Effective Date	NA	NA
Aggregate Function	NA	NA	Count
Aggregate Record Name	NA	NA	Employees Employee Details

5. Select the Employee Details record within the Employees data group and using the same method enter the following extract attribute details.
6. Complete the general fields, as shown in this table:

Field Name	Attribute Entry	Attribute Entry	Attribute Entry
Name	Full Name	Gender	Date of Birth
Start Date	1/1/00	1/1/00	1/1/00
Data Type	Text	Text	Date
Type	Database Item Group	Decoded database item group	Database item group
Database Item Group	Person Full Name	Person Gender	Person Date of Birth

Field Name	Attribute Entry	Attribute Entry	Attribute Entry
Name	Salary	Bonus	Tax rate
Start Date	1/1/00	1/1/00	1/1/00
Data Type	Number	Number	Text

Field Name	Attribute Entry	Attribute Entry	Attribute Entry
Type	Database item group	Record Calculation	Rule
Database Item Group	Assignment Salary Amount	NA	NA
Calculation Expression	NA	Salary * 0.5	NA
Rule	NA	NA	FAST Bank Tax Rule

Defining the Delivery Options

1. Select the Deliver icon and then the Add icon to define the delivery options.
2. Complete the general fields, as shown in this table:

Field	Value	Value
Start Date	1/1/00	1/1/00
End Date	12/31/12	12/31/12
BI Publisher Template	ReportLayout	EFTLayout
Output Type	PDF	EFT
Delivery Type	E-mail	FTP
Delivery Option Name	E-mail to HR	FTP to 3rd Party
Output Name	HeadcountReport	EFTReport

3. Ensure you enter the additional information such as, the server, username, and password for the FTP delivery type.
4. Enter FAST Bank Extract as the reporting category and click Submit.
5. View the extract definition details and ensure the structure is valid in the Validate page by selecting the Validate icon.
6. Select Export XSD to download the XML Schema Definition (.XSD) file for this extract setup. This exported file contains the structure of the extract definition: the data groups, records, and attributes.

Submitting an Extract

An extract definition automatically creates an extract process (payroll flow) with the same name as the extract. The extract process enables you to define an execution sequence of multiple tasks, including before and after tasks.

1. Select the Submit Extracts task and select the FAST Bank Extract process.
2. Select Next.
3. Enter FAST Bank Extract - Jan 2012 as the Payroll Flow (extract process).
4. Enter 1/1/15 as the End Date.
5. Select Next. You can specify interaction details if the task is dependent on other tasks with different extract processes. For example, this task must wait because another task is running.

6. Select Next and review the extract. You can schedule the extract, or run it immediately.
7. Select Submit.
8. Select OK and View Checklist to view the status of the process.
9. Select the View Extract Results task to review the results of the extract run. Search for the FAST Bank Extract process.
10. Select Go to Task for FAST Bank Extract - Jan 2012, click the eyeglasses, and view the report output by selecting the report name.

Related Topics

- [How do I create a BI Publisher template for HCM extract?](#)

Validating Extracts: How It Works

Use the Validate feature in the Manage Extract Definitions task to ensure the data you enter in the Design tab is valid and there are no issues when you submit. After creating an extract, go to the Validate tab and click Validate. If there are any issues in the extract design, validation messages are displayed.

How extracts are validated

When you validate an extract, the application performs validations on the extract design to ensure that:

- The root data group is defined.
- All non-root data groups are linked to the root data group directly or indirectly (for example, through another non-root data group that is linked to the root data group).
- If a sequence of data groups is defined, then the next data group is also defined for processing.
- All Fast Formulas used in the extract exist and are compiled or valid.
- There are no issues detected during BI Publisher validations.

If the validation is successful, the application marks the extract as valid. However, if the validation fails, the application marks the extract as invalid and submitting the extract results in errors.

Viewing Extract Run Details: Explained

Using the View Extract Run Details task, you can review extract run information and troubleshoot extract runs that are unsuccessful or have not produced the expected results. In the Search area, use the various filters available to search the extract or extract run. In the Search Results area, click the extract run that you want to analyze.

In the View Extract Run Details page, you can review the following information:

- Parameters
- Archive details
- Process details
- Changes Only Details
- Delivery options

Parameters

The Parameters tab displays the parameters set for the extract run. For example, Baseline Only, Changes Only, Process Start Date, Process End Date, Process Configuration Group, and so on. You can also export these parameters to an Excel Spreadsheet.

Archive Details

The Archive Details tab displays the number of records that were extracted by the extract run.

Process details

The Process Details tab lists the processes and their hierarchy. You can view the process status, start and finish time, and the elapsed time. You can also download the process log files, which can help you troubleshoot failed or time consuming extract runs.

Extract Delivery Options

The Delivery Options tab is visible only if you have set the delivery option in the Manage Extract Definitions task as WebCenter Content or Inbound Interface. If the extract run is submitted and completed successfully, you can also download the output file.

Changes Only Details

The Changes Only Details tab allows you to review the change in an attribute from the last successful run. This tab is visible only when the extract is run in the Changes Only mode. You can verify if the Changes Only parameter was set in the Parameters tab. Select the attribute and enter the attribute value to check the change in the attribute from the last successful run.

FAQs for Managing Extracts

What's the Simplified interface in HCM Extracts?

The Simplified interface is an easy-to-use graphical interface for defining and designing HCM extracts. You can perform most of the tasks for defining the extract in the Simplified interface, however to enter an effective date, you must use the Desktop interface.

What's an extract data group connection?

Data group connections enable you to define the master-detail or parent-child relationship between the entities. For example, the Employees and Departments data groups are linked with Department ID.

How can I disable an extract?

You can disable an extract by using the Manage Extract Definitions task in the Data Exchange area. Search for and open the extract that you want to disable. Change the status of the extract to Inactive. After disabling an extract, you cannot submit the extract run again. The previous submitted runs of a disabled extract are stored and are available for review.

Why can't I submit an extract run?

You can't submit an extract run when:

- The status of the extract is inactive. An inactive extract indicates that it's disabled. To submit an extract run, the status must be Active. Inactive extracts are not available for submission in the Submit Extracts task.

- The extract is invalid. After creating an extract, if you validate it and there are some validation errors, the extract is marked as invalid. If you try to submit an invalid extract, the process fails and errors are displayed. Resolve the validation errors and validate the extract. Then submit the extract run again.

Why can't I find my inactive extracts?

You can't find the inactive extracts because by default, the search results display active extracts only. To view inactive extracts, use the Manage Extract Definitions task in the Data Exchange work area. In the Search page, select Inactive in the **Status** drop-down list and click **Search**.

If you want the search results to display both active and inactive extracts, select the blank option in the **Status** drop-down list.

Do I have to validate existing extracts?

No. It's not required to validate existing extracts. Extracts that are not validated are marked as Not Yet Validated. You can submit extracts that are not validated. However, if there are issues with the extract design, then errors may occur or the result data may be incorrect. To avoid any errors, ensure that you validate the extract before submitting.

How can I download log files for the extract run processes?

Use the View Extract Results task. Search for the extract and select an extract run. In the Details area, the Process Details tab lists the extract run processes. You can download the log file for any process using the Log column. You can review the log files to identify the records that did not retrieve the expected data and to identify time consuming processes.

How can I verify if a user entity can be used as a root data group?

Search for the user entity in the View User Entity Details UI. If it is a multiple row user entity and does not require any context, you can use this user entity as the root data group.

You can access the View User Entity Details UI:

- From the Data Exchange tasks pane.
- When creating a data group in the Design tab of the Manage Extracts Definitions task.

Can I run a changes only extract with successful runs only?

Yes. Use the Include changes from last successful run option to exclude the extract runs that did not complete successfully when you run a changes only extract. Leave the option unchecked if you want the application to include both the archives from extract runs that completed successfully and those that completed unsuccessfully when you run a changes only extract.

How can I create and run a baseline extract?

Use the Baseline Only parameter in your extract to create a baseline for which you can run all subsequent changes only extracts against. This parameter can save time when running the extract because it doesn't generate an XML file and it doesn't deliver any output. You set the Display option to Yes or Mandatory for the Baseline Only parameter. It creates a full extract in less time and uses less storage in your application.

Can I create an extract and delete the archives?

Yes. When you create a full extract of data, use the Delete Archive parameter to give you the flexibility of deleting archive and XML files at a later date. Set the Display option to Yes or Mandatory for the Delete Archive parameter. The application generates the archive and XML files and delivers them to the destination, for example, WebCenter Content. The Delete Archive parameter provides you with options to delete the archives and XML files after the application delivers the extract data to its destination. After the application delivers the output, for example, to WebCenter Content, you have the option to discard the archive data and XML files, therefore reducing storage consumption by deleting these extract archives.

You also have the flexibility to choose whether you discard all the data an extract generates, for example, use the Delete archive information and generated XML option to remove all unnecessary data. Or you can choose the Delete archive

information only option to remove the archive data and retain the generated XML data. You may want to keep the XML data for reporting purposes or for future references.

Filtering Extracts

Using Extract Modes: Explained

You can create an extract to output data in different modes in the Manage Extract Definitions page. Use modes to extract data that has changed since the previous extract runs. For example, you can extract employee details whenever there is a change in the employee's location. The processing engine generates the current status of the data, compares it with the base-lined data of the previous runs, and identifies the new as well as any modifications. The output from such changes-only extract has the incremental data only.


You can manipulate the output in a changes-only extract to either exclude or include certain attributes, regardless of whether the attribute has changed or not. For example, by selecting an option to exclude an attribute from comparison, you can ensure that that attribute is not compared while identifying changes to include in the generated output of the next extract run. Therefore, even if the attribute has changed since the previous extract run, it will not be included in the output. You can also select an option to always include an attribute in the generated output of an extract run, even if the attribute has not changed.

Extract Modes

To enable different modes in an extract, you include the `CHANGES_ONLY` parameter and set up threading details in the extract definition. While running the extract you can use the `CHANGES_ONLY` parameter to control the extract mode. For example, to find out if the job name has changed on a person's assignment, set up multi-threading database items at assignment level and include the `CHANGES_ONLY` parameter. When you run the extract in the `ATTRIBUTE` mode, the extract compares data from the multi-threading level and outputs the incremental changes. You can link the `PER_EXT_CHANGES_ONLY` lookup or the `ORA_HRY_CHANGES_ONLY` lookup to the `CHANGES_ONLY` parameter so that you can select a mode when you submit the extract.

The following table describes the different extract modes, their lookup values and descriptions.

Mode	Lookup Value	Description
N	All attributes	Includes all data in the extract. A full extract is run which produces the full data output at that point of time. The archived data is utilized as a baseline.
Y	Changed attributes	Compares this extract run with the previous extract runs and by comparing against the baseline (to identify the incremental data), displays the data that has changed only.
ATTRIBUTE	Changed and marked attributes	Includes elements that have changed or marked as mandatory
ATTRIB_OLD	Changed and marked attributes with previous values	Displays elements that have changed or marked as mandatory plus their previous value

Mode	Lookup Value	Description
 Note: You must run the Payroll Interface with the Attrib_ Old mode whenever you use the US ADP PayForce Third-Party Periodic Extract.		
BLOCK_OLD	Changed, marked attributes, previous data under threading group	Displays the following data under threading data group: Changed data Data marked as mandatory Previous values Parent data group values
BLOCK	Changed and marked attributes under threading group	Displays the following data under threading data group: Changed data Data marked as mandatory Parent data group values

Filtering Extracts Using Advanced Filter Criteria: Example

Use advanced filter criteria to filter and extract data using complex SQL queries. You can also filter data using expressions or fast formulas. When using expressions, the available operators may not be sufficient to build complex expressions. You can use fast formulas to build complex filter criteria. However, using advanced filter criteria can improve the performance of the extract run. To use advanced filter criteria, you must have good understanding of the table aliases and writing SQL constructs.

Extracting Employees on Leave

This example explains how to use advanced filter criteria to extract employees who are going on leave in the next seven days. Managers can use this information to find the list of employees who have approved leaves starting in the next seven days.

1. On the Manage HCM Extract Definitions page, create an extract definition and use the Switch Layout button to open the extract in the Professional interface.
2. Select the Data Group link from the navigation tree to open the Data Groups page, and click Create.
3. Select the user entity PER_EXT_SEC_PERSON_UE and click Advanced.
4. In the Query tab, review the user entity tables and aliases that you want to use, and click OK.
5. In the Data Group Filter Criteria area, click Add.
6. Click the Edit icon in the Filter Criteria column.
7. In the Edit Filter Criteria Condition window, create the following expression for retrieving only employees: Extract Person System Person Type = 'EMP'
8. Click Advanced. The application converts the expression in the basic mode to the following SQL construct:
(pptum.system_person_type='EMP')
9. You can now append new SQL construct by using the EXISTS clause. Entering the following SQL construct will extract employees who are going on approved leave in the next seven days:

```
( (pptum.system_person_type='EMP') and EXISTS
(
select 1 from fusion.ANC_PER_ABS_ENTRIES abs
where
abs.person_id = pptum.person_id and abs.start_datetime between
pay_report_utils.get_parameter_value_date('EFFECTIVE_DATE') and
(pay_report_utils.get_parameter_value_date('EFFECTIVE_DATE')+7)
and abs.approval_status_cd = 'APPROVED'
)
)
```

When writing an SQL query, if you want to retrieve any parameter value at run time, use the 'pay_report_utils.get_parameter_value_date('ESS PARAMETER')' function. The ESS Parameter is typically, unless explicitly modified, the capitalized parameter name where any space is replaced by an underscore. For example, the ESS Parameter of Effective Date is EFFECTIVE_DATE.

 **Note:** You cannot use more than 2000 characters to write the SQL query.

10. To validate the SQL construct for any syntactical errors, click Validate.
11. Click OK.
12. To verify if the advanced SQL criteria is applied correctly, run the extract in GMZFT logging mode and check the logs.

FAQs for Filtering HCM Extracts

How can I restrict the records to be extracted?

You can use extract data group criteria to define a set of filtering conditions the application performs on an extract data group. For example, you can use database items in the fast formula to represent the town_or_city and primary_flag columns in the per_addresses table to restrict the data to people living in London only. The extract definition would then exclude people with a primary address of anywhere other than London. You can specify the criteria conditions using an expression or fast formula.

What's a threading database item and what is its connection to the extract data group?

A threading database item is required for implementing the Changes Only feature. The threading database item is a unique ID in the chosen user entity. Generally, for Pay Employee user entity and Assignment user entity it would be DBI with %ASSIGNMENT%ID. For Person user entity, it would be DBI with pattern %PERSON%ID. You can declare one threading database item at the root data group or any child data group level. For example, you declare the threading database item from the location where you need changes only.

What's a conditional action?

A conditional action identifies the action to perform, and optionally, a message based on the outcome of a conditional expression or a predefined fast formula. Conditional actions are applied on the extracted data similar to criteria conditions that are applied prior to extraction. Actions and messages are predefined in lookups, and you can add your own messages by creating new values for the lookup.

When a condition is satisfied, you can use this feature to perform certain predefined actions. For example, you can exclude employees that satisfy a condition, such as all employees from a predefined country. You can also configure this feature to raise a warning when an employee's salary is blank or beyond a specific level.

What's an exclusion rule?

You can't exclude or override a record that doesn't suit your requirements with your own record by using an exclusion rule. The extract process doesn't process excluded records based on the legislative data group.

What is extract data group filtering?

Extract data group filtering filters data extracted in the data group. You can specify to filter data group as an expression or as fast formula, and you can choose one or both. You can build criteria using the available database items, parameters and operators (conditional and logical). The filter criteria will be more efficient as it gets appended to the User Entity SQL at the time of execution. If you can't specify the criteria as an expression, then you can place logic inside a fast formula and select here. Fast formulas return values of Y or N to indicate if you must extract the record or not. If both criteria and formula are specified, then both the conditions are applied.

What are extract data group criteria?

You can specify the filter conditions of what data you want to archive by using the data group criteria. You can specify the filter conditions as an expression or fast formula.

Delivering Extracts

Delivery Options: Explained

You specify delivery options as part of the extract definition. Delivery option parameters specify the output format, the delivery method (email, FTP), and other parameters required for integration with BI Publisher layer.

How Delivery Options Work

HCM Extracts archives the extracted data into result tables and stores it as an XML output in the database. The application transforms the XML output into a formatted output such as HTML, PDF, EDT, or XLS. The formatted output is then delivered through email, fax, FTP, or print, depending on the delivery options you set in the extract definition. You can define delivery options for an extract using a BI Publisher template, with the following delivery file output types: PDF, XLS, XML, DOC, and the following delivery modes: FTP, email, and fax.

Using Delivery Types

The type of delivery you select determines the destination of the extract. Some delivery types require additional information. You can select Documents of Record as the delivery mode to store the output in the database and allow employees to view the output from document of records. An example of a document is a payslip. If the XML output is split and burst as separate files, then you can select the bursting node. For example, if you want all employees to receive an email with their payslip, then set the bursting node to Employee_ID. Select the WebCenter Content delivery type to create extracts with encrypted or non-encrypted data and transmit them to Oracle WebCenter Content. You can then transfer the data manually or using your own scripts to your own server. For more information, see Oracle Fusion File Transfer Automation and Data Security on My Oracle Support.

Defining the BI Publisher Template: Worked Example

This example demonstrates how to set up a BI Publisher template and make it available for you to create a document in the required format, using the extracted data. You access BI Publisher using the Reports and Analytics link from the navigator. When the Reports and Analytics page is accessible, you can browse through the catalog.

FAST bank must send an XML file on their employees and departments to a third party and the HR Manager with employee details grouped by departments as a Headcount Report.

The following table summarizes some key decisions:

Decisions to Consider	In this Example
Why do I have to set up BI Publisher?	You set up BI Publisher to format the extracted XML data into the required format and to deliver the report or business document to the appropriate destination.
How does BI Publisher connect to HCM extracts?	You install the Template Builder for Word plug-in.
Do I need a separate license?	No, BI Publisher is included with Core HR.

Tasks

1. Setting up the Data Model
2. Creating the BI Publisher Report Template
3. Uploading the Report

Setting up the Data Model

1. Select the Reports and Analytics option from the navigator.
2. Select the Browse Catalog icon.
3. Select the New menu option, and under the Published Reporting section, select Report.
4. Select the Use the existing data model option to create a report using the existing data model.
5. In the window that displays, select the data model and select Next.
6. Select the Report Editor option and select Finish to complete the data model setup.
7. Save the report as HR Data Report in the following folder: /Shared Folders/Custom/Human Capital Management/

Creating the BI Publisher Report Template

1. Install the Template Builder for Word plug-in from the BI Publisher Enterprise Home page.
This plug-in provides sample documents, demos, templates and Template Viewer. Use these samples to help you understand the concepts and to create templates such as EFT and RTF to view the formatted XML output.
2. Create a BI Publisher template using the Export XSD option in the extract execution tree.
3. Open MS Word and locate the Add-Ins tabbed region to view the Oracle BI Publisher option.
4. Select the Load XML Schema option. After the XML schema has loaded, you can arrange the fields and alter the layout, if required.

Uploading the Report

1. Select the Upload option from the Upload or Generate Layout region and enter the following information:

Parameters	Details
Layout Name	RTF Layout
Template File	Select the file name of the RTF template saved on your local drive.
Type	RTF Template

Parameters	Details
Locale	English (United States)

2. Save the BI report in the following location: /Shared Folders/Custom/Human Capital Management/.

This BI Publisher report template is now ready for you to select in the Delivery Options section when creating the extract definition.

Related Topics

- [How do I create a BI Publisher template for HCM extract?](#)

Delivering Encrypted Data Using HCM Extracts: Explained

You can deliver encrypted files using HCM Extracts to Oracle WebCenter Content. You use the WebCenter Content delivery type in the Deliver page of the Manage Extract Definitions task to create an encrypted extract. HCM Extract transfers the encrypted file to Oracle WebCenter Content. You can then transfer the files manually, or write scripts to perform the transfer. By selecting the WebCenter Content delivery type you can enter additional details to ensure the extract is identifiable in Oracle WebCenter Content. For example, specify the integration name, select whether you want to encrypt the file using the encryption mode, and a file name for your reference. This additional setup ensures the document has the correct encryption mode and is stored in Oracle WebCenter Content with a specific file name. For more information, see Oracle Fusion File Transfer Automation and Data Security on My Oracle Support.

FAQs for Delivering HCM Extracts

Can I use Oracle Fusion Transactional Business Intelligence (OTBI) with HCM Extracts?

Yes. You have two options:

HCM Extracts can extract the data and produce an output using CSV, XML, or PDF. OTBI can then accept a data source in Excel or XML format.

BI Publisher can also accept a data source in Excel or XML format. HCM Extracts has integration with BI Publisher.

What's BI Publisher and how does it work with HCM extracts?

BI Publisher is a set of tools you use to create highly-formatted reports based on data models. With BI Publisher, you can:

- Author, manage, and deliver documents
- Create interactive management reports
- Create highly-formatted, customer-facing documents
- Create government documents
- Create electronic funds transfer (EFT) documents

BI Publisher transforms the extracted data from the database and presents that data into a report.

How can I reduce the size of my extract output?

In the Delivery tab, set the Compress option as Yes. You can use the compress feature with the WebCenter Content delivery option only. This feature extracts a compressed output, which helps to speed up large file transfers, uploading, and downloading. Compressing the extract output uses less disk space too.

Does a delivery option have to complete for an extract to run successfully?

No. Use the Required option in the Extract Delivery Option table to specify which delivery options you need to run successfully in order for the extract run to succeed. For example, use the Required option for critical delivery options such as payroll bursting. If this delivery option fails, then the extract run will also fail because it is dependent on the delivery option completing successfully.

You can also use the Required option for any delivery options you deem as not critical for the extract to run successfully. For example, deselect the Required option for a delivery option such as an email notification to yourself. If this delivery option fails, then the extract run will complete successfully because it is not dependent on the delivery option.

26 Flow Patterns

Flow Patterns for Extracts and Reports: Overview

A flow pattern exists for each extract or report process. The flow pattern defines submission parameters, flow tasks, and task sequence. Flow patterns for extracts and reports typically contain a single flow task. You can edit them to add tasks in the Payroll Checklist work area or by using the Refine Extract task in the Data Exchange work area. Each flow pattern you create must be associated with a security profile.

Checklist and Flow Tasks: Explained

A flow can consist of one or more tasks. The flow pattern determines the sequence of tasks executed in a flow. Submitting a flow from the Data Exchange or payroll work areas generates a checklist. Use the flow's checklist to monitor and manage the tasks included in the flow.

Depending on the flow pattern, the checklist might include:

- Automatic tasks, such as extracts, reports, and processes
- Manual tasks, such as verification tasks required to complete a flow

Working with Checklists

Use the checklist while working with flows to perform the following activities:

- Monitor the status of the flow tasks
- Manage the flow tasks, such as reassigning tasks, marking tasks completed, and performing corrective actions
- View task details, such as a list of records processed by the flow

For payroll, while working on a task in the flow, you can remain in the Payroll Checklist work area or go to a related work area that includes tasks in the regional area. For example, while reviewing the results for the Calculate Payroll task, you might go to the Payroll Calculation work area to review the person's calculation card or element entries.

Related Topics

- [Monitoring the Status of Flow Tasks: Explained](#)

Editing Flow Tasks: Points to Consider

Edit flow patterns you create or copy. This topic suggests points to keep in mind when you add, delete, or move a task in a flow pattern.

Editing Tasks

Refer to the following table for examples of edits to flow pattern tasks. You perform these edits on the Task Sequence tab of the Manage Payroll Flow Patterns page. For payroll flow patterns, use the Manage Payroll Flow Patterns task in the Payroll Checklist work area. For extract flow patterns, use the Refine Extracts task in the Data Exchange work area.

Edits	Impact	Examples
Add a task	<p>You add a task to position it as the last task in the activity or task group. Update the task sequence.</p> <p>If you repeat a task, rename it to make clear its purpose on the checklist.</p>	You add a manual verification task after each report. You rename each task with the report name.
Delete a task	When you delete a task you may impact subsequent tasks in the flow that depend on its results. Review the subsequent tasks.	You delete a task. The Parameter Basis of the next task is Bind to Task and its Basis Value is the value of the deleted task. You update the Parameter Basis of the subsequent task as required, for example, to Bind to Flow.
Move a task to a different activity	The activity determines the work areas where you can submit the flow patterns you define, and controls how the checklist displays.	You move a task in a payroll flow pattern for a report from the Payments activity to the Statutory activity. The flow owner can view the report results from the Payroll Checklist or Regulatory and Tax Reporting work areas, but not the Payment Distribution work area.
Reorder the list of tasks displayed in a checklist	The sequence specified for the task further determines the task order within an activity on the checklist.	<p>You decide to flatten the checklist sequence to group all the tasks within a single activity.</p> <ol style="list-style-type: none"> 1. On the Tasks page, you confirm that each task belongs to the same activity and task group. 2. You edit each task, specifying a value in the Sequence column on the Edit Task Details Owners and Details page. <p>The lowest number is used for the first task in the checklist. For example, you might specify a sequence of 10 for the payroll calculation task and 20 for the prepayments calculation task.</p>

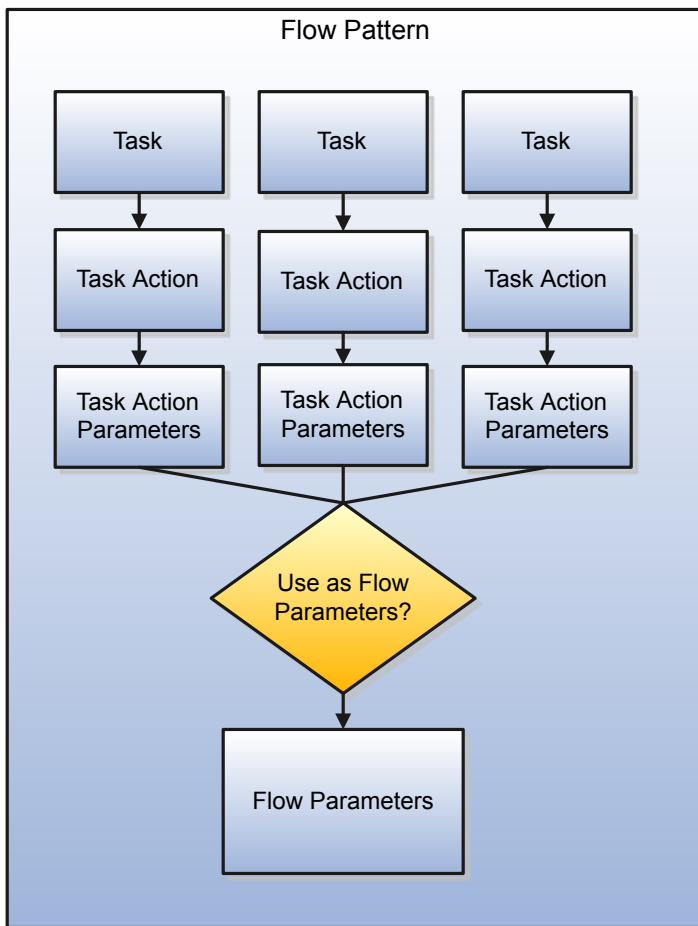
Related Topics

- [Editing a Flow Pattern: Worked Example](#)
- [Sequencing Rules for Flows and Locked Tasks: Explained](#)

Flow Pattern Parameters: Explained

Each task in a flow pattern supports task actions, such as submit, roll back, mark for retry, retry, and view. Task action parameters control how the application processes a task and how the task relates to other tasks in the flow pattern. Flow parameters, a subset of task parameters, supply the information required to successfully complete the tasks in the flow pattern.

The following figure shows the relationship of the tasks, task parameters, and flow parameters in a flow pattern.



When you create a flow pattern, you review and edit the task parameters for the Submit and Initialize task actions. After submitting the flow pattern, you can edit the parameters for the remaining task actions, such as Mark for Retry, Retry, and Roll Back. The parameter details you can edit include:

- Display and display format
- Lookups and value sets
- Usage
- Sequence

- Parameter Basis and Basis Value

Display and Display Format

Display parameters control the format and availability of the flow parameter, as shown in the following table.

Parameters	Purpose
Display	Determines whether the parameter displays when submitting the flow
Display Format	Identifies the type of data displayed, such as a date or text, or choice list

Display parameters work with other parameters, such as Parameter Basis and Basis Value. For example, most task action parameters don't display the Request parameter because the application obtains the value for this parameter from the context.

Lookups and Value Sets

Use lookups and value sets for descriptive flexfields to control and validate the data used in the payroll flow pattern.

The following table describes which parameter basis to use for the different methods for obtaining the lookup value.

Lookup Value	Parameter Basis
Entered when submitting a flow	Bind to Flow
Derived by the application from existing tables, such as the value for the payroll statutory unit	Bind to Flow Task or Context Binding
Derived by application from a Post SQL process	Post SQL Bind

Usage

A parameter can receive information (input) or generate information (output) that subsequent tasks can use as input. For example, for the Calculate Payroll task, the Payroll Process parameter for the Submit task action generates an output value for the payroll action ID, which the Retry task action uses as input.

The following table describes the typical settings for a parameter whose usage is output.

Parameter Option	Setting
Display	No
Parameter Basis	<ul style="list-style-type: none">• Bind to Flow, the application derives the value from the flow parameter and the updates the flow parameters table with the output value• If you select no value, the output value results from the task's output

Sequence

Control the order in which the application processes and displays the parameters by specifying the sequence. Sequence numbers provide default logic for the application to derive the order in which to process the parameters. For example, if you have two lookups and the values of the second lookup depends on the first lookup, set the first lookup to a lower sequence number than the second one

Parameter Basis and Basis Value

The Parameter Basis controls how the application derives the value for the parameter. The Basis Value further specifies the value the application uses for the parameter.

The following table lists the Parameter Basis options and gives examples of when you might select them.

Parameter Basis	Description	Basis Value Available	Example
Use Specified Value	Assigns a specific value to the parameter.	Text is entered as constant or value entered by the person who submits the flow.	Specify a constant if the value is the same for all tasks, such as the payroll statutory unit.
Bind to Context	Derives the value from the context of the current flow instance or the task instance of the flow pattern.	Select flow, task, or the Request. The application will automatically generate the parameter value.	<p>If the task includes a Request parameter, bind it to the flow context. Tasks in the flow reference this task using the Request ID generated by the application.</p> <p>Bind the legislative data group parameter to a task parameter that supplies the legislative data group.</p> <p>For example, the legislative data group for prepayments uses the payroll as context, because it's already associated with the legislative data group.</p>
Bind to Flow Parameter	Derives the value from one of the flow parameter values.	Application will automatically derive the parameter value.	Bind a parameter to the flow that several tasks share to avoid multiple occurrences of the same parameter.
Bind to Flow Task Parameter	Binds the value to the output of the previous task.	Select a value from the previous task's parameters.	Bind a parameter to a task, such as Retry corrective action, so that when the flow owner resubmits the task to retry it, the application uses the output of the Submit task parameter.
Bind to Task Parameter	Resolves the value for the task parameter.	Select a value from the current task's parameters.	Bind a parameter to the task if several tasks share a parameter, such as a start date, but one task requires a different date.

Parameter Basis	Description	Basis Value Available	Example
No value specified	Stops the application from generating a parameter value when the task executes.	Application will generate a blank value.	Not applicable
Post SQL Bind	Calculates the parameter but doesn't display it on the user interface.	SQL statement will calculate the parameter value.	Bind a parameter using the Post SQL bind to generate data. For example, use a post SQL process to generate the process date from the payroll period and payroll parameters.
SQL Bind	Calculates and displays value on the user interface.	SQL statement will calculate the parameter value.	Bind a parameter using SQL For example use SQL Bind to calculate the payment type parameter for the Generate Check Payment task. The application obtains the payment type ID for the check payment record. Use SQL Bind to prompt the task owner to enter a reason for a corrective action, such as a QuickPay.

Related Topics

- [Creating a Flow Pattern to Reissue a Check: Worked Example](#)
- [Can I skip the flow parameters for a single-task payroll flow pattern?](#)


Flow Task Start and Due Dates: Critical Choices

Specify duration dates and notification options in the payroll flow pattern to give flow owners adequate time before a task starts to prepare and before a task is due to address any issues.

Task Start and Due Dates

Specify the following duration dates on the Tasks page of the Manage Payroll Flow Patterns page:

- Start date, the date the task owner should start the task

 **Note:** The start date applies to notifications only. You schedule when a flow starts on the Scheduling page when you submit the flow.

- Due date, the date the task owner should complete the task

To specify duration dates:

1. Select the flow parameter date to use as the basis for the duration date

2. Optionally, offset the date by specifying a plus or minus value depending on whether the date falls before or after the duration date.

Notifications

Select notifications to send error and warning messages, and to inform the task owner when a task starts or ends. The receipt of notifications depends on the duration dates and their offsets.

1. Specify the notifications the task owner receives.
2. Optionally, specify the number of days before the application automatically deletes a notification from storage.
- Use the Manage Payroll Process Configuration task in the Setup and Maintenance work area to complete the Notification Expiration Offset parameter.

Related Topics

- Scheduling Flows: Explained

Editing Flow Patterns: Examples

Review these scenarios to better understand how to edit flow patterns to meet the requirements of your enterprise. Use the Manage Payroll Flow Pattern task in the Payroll Checklist work area or the Refine Extracts task in the Data Exchange work area for the following scenarios:

- Updating a parameter to use a specified value
- Supplying a reason for a corrective action
- Adding tasks and reordering the task sequence
- Automatically incrementing dates in a scheduled extract

Updating a Parameter to Use a Specified Value

Your payrolls use a single process configuration group named InFusion UK Consolidation Group. You want to specify a constant for the configuration group task action parameter and hide the parameter to avoid data entry mistakes. You perform the following steps.

1. Query the flow pattern you defined for the payroll cycle.
2. On the Parameters tab of the Manage Payroll Flow Pattern page, edit the Process Configuration Group parameter.

You enter the values shown on the following table.

Parameter Detail	Value
Display	No
Display Format	Text
Lookup	No value

Parameter Detail	Value
Usage	Input Parameter
Parameter Basis	Constant Bind
Basis Value	InFusion UK Configuration Group

Supplying a Reason for a Corrective Action

Your enterprise typically issues electronic funds transfer payments. You defined a flow pattern to issue check payments and to verify them. You want to track the reason managers issue checks, so you add a flow parameter to capture that information.

1. Query the payments flow pattern you defined.
2. On the Parameters tab of the Manage Payroll Flow Pattern page, Select and Add the Reason parameter to include it as a flow submission parameter, and complete the parameter details as shown in the following table.

Parameter Detail	Value
Display	Yes
Display Format	Text
Lookup	No value
Usage	Input Parameter
Parameter Basis	Context Binding
Basis Value	Payroll Flow

Adding Tasks and Reordering the Task Sequence

Your flow pattern includes a Calculate Gross Earnings process and the Element Results Register Report. You decide to add two extract reports that run concurrently, followed by a verification task, and to simplify the checklist so that it is a single list. You perform the following steps:

1. From the Data Exchange work area, select the Refine Extracts task.
2. On the Refine HCM Extracts page, query the flow pattern.
3. On the Tasks tab of the Manage Payroll Flow Patterns page:
 - a. Add the first extract report, specifying the same Activity and Task Group as the Calculate Gross Earnings.
 - b. Add the second extract report, specifying the same Activity and Task Group as the Calculate Cross Earnings.

- c. Add a manual verification task, specifying the same Activity and Task Group as the Calculate Cross Earnings.
4. Edit each task, specifying a sequence number on the Edit Task Details Owners and Details page.
The lowest number is used for the first task in the checklist. For example, you might specify a sequence of 10 for the Calculate Gross Earnings task, 20 for the first extract report, 30 for the second extract report, and 40 for the manual verification task.
5. On the Tasks Sequence tab, reorder the sequence of reporting tasks so that the two extract reports run concurrently.

Task	Following Task
Start Flow	Calculate Gross Earnings
Calculate Gross Earnings	First extract report
Calculate Gross Earnings	Second extract report
First extract report	Manual verification task
Second extract report	Manual verification task
Manual verification task	End Flow

Automatically Incrementing Dates in Scheduled Extract

You create a flow pattern to extract weekly payroll data that requires the user to enter a process date parameter. You schedule the extract to run weekly. The application evaluates the flow parameters at the time of submission, and the task parameters when the task starts executing. To automatically increment the date field, so that the application derives the dates from the defaulted date parameter, you edit the task parameters.

You use the Refine Extracts task from the Data Exchange work area, or the manage Flow Patterns task from the checklist work area. You edit the task parameters on the task's Basic Information page by performing the following actions:

1. Select the Process Date parameter.
2. Select Context binding from the Parameter Basis field.
3. Select System Date from the Basis Value field.

Related Topics

- [Editing a Flow Pattern: Worked Example](#)

Adding a BI Publisher Report to a Flow: Procedure

Add single or multiple BI Publisher reports to your copied or user-defined flow pattern. When you submit the flow, the report automatically generates an output file that you can view. The output file is based on the template used for the BI Publisher report, such as an html template. The Run BI Publisher Report task belongs to the Statutory activities in the flow pattern.

Adding Your Report to a Flow

Follow these steps to add the task to add your BI Publisher report to an existing extract flow.

1. Use the Manage Flow Patterns task in the Payroll Checklist work area or the Refine Extracts task in the Data Exchange work area or the area.
2. Search and select the flow that you will customize.
3. On the Manage Flow Pattern page, on the Tasks tab, click the **Edit** button
4. Click the **Select and Add** button on the menu bar. In the Search Tasks dialog, search for and select Run BI Publisher Report. Click the **Done** button.
5. On the Task Sequence page, confirm the sequence is correct.
6. On the Flow Parameters page, add a required parameter for the first argument of the BI Publisher report.

The flow parameters map to the BI Publisher arguments. For example, if your report is based on a SQL query, the first argument is the first bind variable of a SQL query data model.

 **Tip:** To easily determine the sequence of arguments, view the list of parameters for the generated report in BI Publisher.

7. Optionally, rename the parameter to a more meaningful name.
8. On the Task parameters page, in the Parameter Details section, complete the following steps:
 - a. Confirm that the Parameter Basis for the First Argument value is Bind to Flow.
 - b. Specify a value for the Report Name and Report Path parameters.
For example, if the BI Publisher data model is saved to the Custom folder in Shared Folders you would specify `/Custom/yourBIreport.xdo`.
 - c. Specify values for other arguments if required.
9. Review the flow and submit it.

Related Topics

- [Creating a Report to View Batch Line Errors: Worked Example](#)

Creating a Flow within a Flow: Worked Example

This example describes how to copy the Transfer Batch flow and modify it to include a custom flow pattern you created. In this example, the custom flow you add submits a report to check for any batch line errors after transferring a batch. If the transfer fails, you can skip the transfer process or mark it as complete, and then view the report for error details.

Summary of Tasks

This worked example includes details for the following tasks:

1. Creating the parent flow pattern
2. Adding the report flow to the parent flow
3. Testing the flow

Creating the Parent Flow Pattern

1. In the Payroll Checklist work area, select the **Manage Payroll Flow Patterns** task.
2. Search for and select the row for **Transfer Batch**, and then click the **Copy** icon.
3. Enter the name of the new flow pattern, such as **Transfer Batch with Error Report**.
4. Enter a description, such as "Transfer a batch and view any batch line errors that occurred." and then click **Save and Close**.
5. Search for and select the Transfer Batch and Error Report flow pattern, and then click **Edit**.
6. Add the parameter that derives batch name from the batch ID as follows:
 - a. On the Parameters tab, click **Add**.
 - b. Select the added row and click **Edit**.
 - c. Add values as shown in this table.

Field	Value
Flow Parameter	Batch Name
Use for Searches	No
Display	No
Display Format	Text
Sequence	3
Usage	Input parameter
Parameter Basis	Post SQL Bind
Basis Value	select batch_name from pay_batch_headers where batch_id = :BATCH

7. Click **Save**.

Adding the Report Flow to the Parent Flow


1. On the Tasks tab, click **Select and Add**.
2. In the Search window, search for and select **Submit Another Flow**, and then click **Done**.
3. In the row for Submit Another Flow, click the **Edit** icon in the menu bar and set the values as shown in this table.

Field	Value
Task	Run Batch Lines Report
Activity	Statutory

Field	Value
Task Group	Reporting
Description	Submit the batch lines error report for the specified batch.

4. Edit task parameters as follows:

- a. In the row for Run Batch Lines Report, click **Edit Task**.
- b. Configure the predefined task parameters as shown in this table.

Parameter	Parameter Basis	Basis Value
Flow Name	Constant Bind	The name of the flow, for example Batch Lines Report.
 Note: This value is case-sensitive. Enter the name exactly.		
From Flow Instance ID	Context Binding	Payroll flow
From Flow Task Instance ID	Context Binding	Payroll task
Use to Calculate Results	Constant Bind	Y
Parameter Name 1	Constant Bind	Batch Name
Parameter Value 1	Bind To Flow	Batch Name

- c. Click **Next**, and optionally complete the owner and checklist information.
 - d. Click **Next**, and optionally complete the duration and notification information.
 - e. Click **Submit**.
5. Edit the task sequence as follows:
- a. On the Task Sequence tab, edit the following two rows as shown in this table

Start Flow	Following Task
Transfer Batch	Run Batch Lines Report
Run Batch Lines Report	End Flow

- b. Click **Submit**.

Testing the Flow

1. Create and save a test batch that should cause an error.

Alternatively, you can search for an existing batch that was transferred with errors using this SQL query:

```
select * from pay_batch_headers where batch_status = 'E';
```

2. On the Submit a Process or Report page, select a legislative data group.
3. Select the **Transfer Batch with Error Report** task, and then click **Next**.
4. Enter a unique name for the current flow instance.
5. Enter the name of the batch with errors that you saved or queried, and then click **Submit**.
6. Click **OK and View Checklist**, and then click the **Refresh** icon until the Transfer Batch task shows as in progress with error.
7. View the report in the flow as follows:
 - a. Select the row with the Transfer Batch task, and then select **Skip Task** in the Actions menu.
 - b. In the row for Run Batch Lines Report, click **Go to Task**.
 - c. In the Processes and Results section, click the name of the report.
 - d. In the row for Run BI Publisher Report, click **Go to Task**.
 - e. On the Process and Reports tab, click **View Results**.
 - f. Click the PDF file name to open the report.

Related Topics

- [Creating a Report to View Batch Line Errors: Worked Example](#)

FAQs for Flow Patterns

What happens if I don't enter a task owner in a flow pattern?

The person who submits the flow becomes the flow owner and the task owner. The person's security privileges determine whether the person can submit the flow.

How can I rearrange tasks in a flow pattern?

Edit the task sequence by selecting a different task in the Following Task column. Every flow pattern begins with a Start Flow task, which does not belong to an Activity or Task Group, and concludes with an End Flow task.

When creating a flow, use the Manage Payroll Flow Pattern task in the Payroll Checklist work area or the Refine Extracts task in the Data Exchange work area. Rearrange tasks on the Task Sequence page. When editing a flow, select the task and edit its sequence on the Create Flow Pattern: Basic Information page. When you submit a flow, processes in the flow use and build upon the results of previous processes. To maintain data integrity, ensure the sequenced tasks follow a consecutive order.

Related Topics

- [Editing a Flow Pattern: Worked Example](#)
- [Sequencing Rules for Flows and Locked Tasks: Explained](#)

What happens if I change a due date for a task in a flow?

It doesn't affect the status or progress information for the flow displayed on the checklist.

You must complete a flow task before beginning the next task. Before updating a due date beyond the next task's start date, consider whether you have adequate time to perform the next task.

Only flow owners and task owners can update the due date.

Why don't the duration dates in the flow pattern display?

The start and end dates and their offsets display after you complete the flow parameter dates. Use the Manage Payroll Flow Patterns task in the Payroll Checklist work area or the Refine Extracts task in the Data Exchange work area. Enter the flow parameters on the Parameters page, and then return to the Tasks page to enter the duration dates.

If your flow pattern doesn't specify dates as flow parameters, the duration list of values is blank. Change the values for the Duration list by displaying the date parameters for tasks in your flow pattern.

How can I improve performance and troubleshoot flows?

Add parameters to a payroll process configuration group to optimize performance and troubleshoot your payroll processes. To process large volumes of records, use the Threads and Chunk Size parameters. To troubleshoot processes, add the Logging Category or Formula Execution Logging parameters to a configuration group and rerun the process using that configuration group. Using these parameters enables you to investigate formula code problems.

Related Topics

- [Payroll Process Configuration Groups: Explained](#)
- [Payroll Process Configuration Parameters](#)

How can I run tasks concurrently in a flow?

Use the Manage Payroll Flow Patterns task in the Payroll Checklist. Search for and open your flow pattern. Click the Task Sequence tab. You specify that each concurrent task follows the same previous task.

For example, you create a flow pattern with a payroll calculation task and two reports. The flow ends when both reports complete. For the first row, you specify the Start Flow task with the Calculate Payroll as the following task. For the second row, you specify Calculate Payroll as the task, and the first report as the following task. For the third row, you specify Calculate Payroll as the task, and the second report as the following task. For the fourth row, you specify the first report as the task and End Flow as the following task. For the last row, you specify the second report as the task and End Flow as the following task.

Related Topics

- [Editing a Flow Pattern: Worked Example](#)

27 Benefits Data Extract

Extracting Benefits Data Using HCM Extracts: Points to Consider

You can extract worker enrollment results, designated dependents and beneficiaries, and corresponding rates and coverages. You create an extract definition using the Manage Extract Definitions task in the Data Exchange work area.

Consider these aspects before you create an extract definition for benefits data:

- Extract type
- Data groups
- Extract record sequence

Extract Type

When you create an extract definition, ensure that you select Benefits Carrier from the Extract Type list.

Data Groups

Create a PER_EXT_SEC_PERSON_UE (Person) data group and select it as the root data group. The following benefits data groups are available:

- BEN_EXT_ENRT_RSLT_UE (Enrollment Results)
- BEN_EXT_ENRT_DPNT_UE (Dependents)
- BEN_EXT_ENRT_BNF_UE (Beneficiaries)
- BEN_EXT_ENRT_RTCVG_UE (Rates and Coverages)

Extract Record Sequence

After you create the data groups, ensure that you sequence the data groups to identify which data group the application processes next:

- Person
- Dependents
- Beneficiaries
- Enrollments
- Rates and coverages

Related Topics

- [Defining an Extract: Worked Example](#)

Configuring for Plan Carriers: Procedure

This topic covers how to set up benefit plan carriers and offerings so that you can then generate and transmit enrollment data extracts. You extract benefits enrollment information into a single XML file for each benefits carrier.

To configure the benefits data extract, you complete these tasks in the Plan Configuration work area:

1. Create the plan carrier and configure extract settings.
2. Add the benefits extract plan type name.
3. Add the benefits extract plan code.
4. Add the benefits extract option name.

Creating Plan Carrier

Set up carrier data and configure extract options in the Plan Configuration work area that apply whenever you run the extract for that carrier.

1. In the Tasks panel drawer, click **Manage Plan Carriers** to open the Manage Plan Carriers page. On the Mapping tab, you can view the mapping of lookup codes to the lookup value that you transmit to carriers. Edits to mapping values affect all plan carriers that use the lookup.
2. On the Search Results toolbar, click the **Create** button.
3. Enter the carrier information. In the **Active** field, select **Active**.
4. Enter the extract options.

The following table provides comments to help you with your extract field entries and selections.

Field	Comments
Extract Type	<p>Select whether to run a full extract or extract only the changes since the previous extract.</p> <p>Generally, you run a full extract after an enrollment period closes and enrollments are completed. You run subsequent extracts on a periodic or scheduled basis, in either full or changes only mode.</p>
Output File Name	Obtain the value that you enter here from the extract file recipient.
Processing Frequency	A common practice is to schedule your extracts to run after your regular payroll runs. You might want to set the processing frequency accordingly.
Processing Type	<p>By default, all extracts have the same format, regardless of which carrier receives the extract. You can contract with Oracle's partner, BenefitsXML, to have a carrier's extract data transformed and delivered to the carrier, according to its specifications. For more information about BenefitsXML, see http://www.benefitsxml.com.</p> <p>If you don't use this partner, you can transform and deliver the extract data file directly to each of your plan carriers, according to their specifications.</p>
Upload Custom Layout	<p>This button is available if you select the Custom layout processing type.</p> <ul style="list-style-type: none"> ○ Specify the layout for the individual carrier. For details, see the Benefits Extract: Custom Layout topic

5. Enter the file transfer details, which you obtained from the extract file recipient. You can transmit extract files directly to the recipient. Alternatively, you can transmit extract files to the Oracle cloud, using its file transfer details, and have your recipient download its extract from there.

Field	Comments
Host	The name of the server to which you transmit the recipient's extract files.
Remote Folder	Path on the host to the location where your transmission places the extract file.
User Name	Part of the sign-in credentials required for your transmission to access the host.
Password	Part of the sign-in credentials required for your transmission to access the host.

6. Click **Save and Close** to return to the Manage Plan Carriers page.

Adding Benefits Extract Plan Type Name

Complete these steps in the Plan Configuration work area to identify each type of plan included in the extract file for the recipient.

1. In the Tasks panel drawer, click **Manage Plan Types** to open the Manage Plan Types page.
2. Search for and click the plan type, for example, Medical.
3. On the Plan Type Definition section **Actions** menu, select **Update**.
4. Enter the valid benefits extract plan type name, for example, Health.
5. Click **Save and Close** to return to the Manage Plan Types page.

Adding Benefits Extract Plan Code

Complete these steps in the Plan Configuration work area to link the plan to the extract file recipient.

1. In the Tasks panel drawer, click **Manage Benefits Plan Details** to open the Manage Benefits Plan Details page.
2. Search for and click the plan that you want to link to the extract file recipient.
3. Click **Next** to open the Edit Plan Additional Configuration page.
4. On the Configuration Details section **Actions** menu, select **Update**.
5. Enter the benefits extract plan code, which the extract file recipient provided to you for this specific plan.
6. Click **Save and Close** to return to the Plans tab.

Adding Benefits Extract Option Name

Complete these steps in the Plan Configuration work area to identify each option included in the extract file for the recipient.

1. In the Tasks panel drawer, click **Manage Benefit Options** to open the Manage Benefit Options page.
2. Search for and click the participant option, for example, Participant Only.
3. On the Basic Details section **Actions** menu, select **Update**.
4. Enter the valid benefits extract option name, for example, Employee Only.
5. Click **Save and Close** to return to the Manage Benefit Options page.

Next Steps

After you complete the previous tasks, you are ready to generate and transmit the extract data. The details of this process are covered in the Generating and Transmitting Benefits Data Extract for Plan Carriers: Procedure topic.

Valid Extract Names: Explained

Oracle partner BenefiX provides valid extract names that you can enter for benefits plan types and options. You can extend these lists of values, as required.

Benefit Extract Plan Type Names

Valid benefits extract names for plan types:

- 24 Care
- Dental
- Dental Capitation
- Exclusive Provider Organization
- Health
- Health Maintenance Organization
- Hearing
- Long Term Care
- Long Term Disability
- Mail Order Drug
- Major Medical
- Medicare Risk
- Mental Health
- Point of Service
- Preferred Provider Organization
- Prescription Drug
- Preventative Care
- Short Term Disability
- Utilization Review
- Vision

Benefits Extract Option Names


Valid benefits extract names for options

- Children Only
- Dependents Only

- Employee and Children
- Employee and Five or More Dependents
- Employee and Four or More Dependents
- Employee and One Dependent
- Employee and One or More Dependents
- Employee and Spouse
- Employee and Three Dependents
- Employee and Three or More Dependents
- Employee and Two Dependents
- Employee and Two or More Dependents
- Employee Only
- Family
- Individual
- Not Applicable
- Spouse and Children
- Spouse Only
- Two Party
- Employee and Domestic Partner
- Domestic Partner and Children
- Domestic Partner Only
- Employee and Spouse or Domestic Partner
- Child or Children of a Domestic Partner

Generating and Transmitting Extracts: Procedure

You can extract benefits enrollment information into a single XML file for each benefits carrier and transmit it to the carrier. This topic covers how to generate and transmit the extract file.

 **Note:** If a carrier provides more than one plan, the single extract contains information related to all plans provided by that carrier. Example: Four different carriers provide ten plans. You run four separate extracts, one for each carrier.

To generate and view the extract, you complete these tasks in the Evaluation and Reporting work area.

1. Submit the extract request.
2. View and transmit the extract details.

Prerequisites

You must first configure your plan carriers, or extract recipients, and add the relevant extract data to the appropriate plan types, plans, and options. The details of this process are covered in the Configuring Benefits Data Extract for Plan Carriers: Procedure topic.

Submitting Extract Request

To run and monitor extracts, in the Evaluation and Reporting work area:

1. In the Tasks panel drawer, click **Extract Benefits Data** to open the Extract Benefits Data page.
2. On the Search Results toolbar, click **Submit**.
3. Enter the extract request options.

Field	Comments
Extract Type	<p>For a particular carrier, you can select whether to run a full extract or extract only the changes since you ran the previous extract.</p> <ul style="list-style-type: none"> ○ Generally, you run a full extract after an enrollment period closes and enrollments are completed. ○ You run subsequent extracts on a periodic or scheduled basis, in either full or changes only mode. Common practice is to schedule your extracts to run after your regular payroll runs.
Transmit	You can transmit the extract as part of the extract request, or after the requested extract completes and before or after you view the extract details.

4. Click **Submit** to submit your process and return to the Extract Benefits Data page.

Viewing and Transmitting Extract Details

You can view, query, and download extracted records for a specific extract run after it completes. You can also transmit the extract after the requested extract completes and before or after you view the extract details.

1. In the Search Results section of the Extract Benefits Data page, click the **Request ID** for the most recent extract request for the plan carrier.
2. Review, query by example, and download to a spreadsheet the extracted data, as appropriate.
3. Click **Done** to return to the Extract Benefits Data page.
4. In the Search Results section, click the **Transmit** icon button for the most recent extract request.

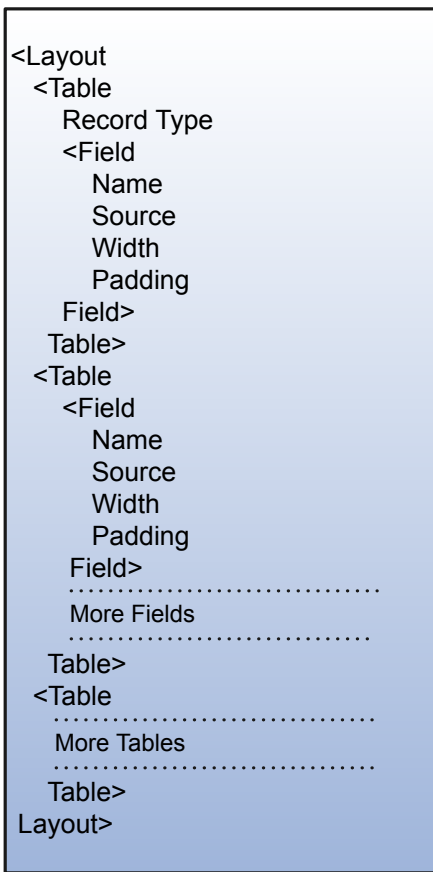
Custom Layout

An implementor or developer can create a custom layout to transform the format of extracted benefits enrollment data to match the specifications of a particular carrier. This topic provides:

- Descriptions of the tags that you require to create the custom layout
- Table aliases and a sample custom layout

The custom layout becomes the default layout for the plan carrier after you upload it to the plan carrier's extract options.

This figure illustrates the structure of the XML tags in the custom layout.



The data source for a field on the custom layout can be a:

- Column on the benefits extract staging tables
- Column on one of the other tables listed in the Source tag description
- Constant into which you enter the exact value

Tip: To identify table column names, you can use the data model query builder in Oracle BI Publisher. Search for the table name and view the columns.

XML Tag Descriptions

This section describes each XML tag and lists its attributes, elements (subtags), and parent tags.

Layout

Description: Root tag.

Attributes: None

Elements (Subtags)	Parent Tag
Table	None

Table

Description: Specifies the database table from which to extract the data.


Attributes: 1. tableName: Supported values = {BEN_EXTRACT_REQ_DETAILS,BEN_EXTRACT_REQUEST,DUAL,PER_ALL_PEOPLE_F,PER_PERSONS,PER_ALL_ASSIGNMENTS_M,PER_PEOPLE_LEGISLATIVE_F,BEN_PL_F,BEN_PL_TYP

Elements (Subtags)	Parent Tag
1. RecordType	Layout
2. Field: See Field tag description	

Record Type

Description: Specifies how to delimit or lay out the data in the extract file.

Attributes: Supported values: {FIXEDWIDTH, CSV}

 **Note:** Anything other than CSV is delimited as FIXEDWIDTH by default.

Elements (Subtags)	Parent Tag
None	Table

Field

Description: Corresponds to one column in the extracted document. Source the text in this column from a database table, an SQL function, or a constant.

Attributes: None

Elements (Subtags)	Parent Tag
1. Name	Table
2. Source	
3. Width	
4. Padding	

Name

Description: Name of the field

Attributes: None

Elements (Subtags)	Parent Tag
None	Field

Source

Description: Specifies the source of data for the current field.

- If the source is a table, the value passed is the column name.
- If multiple tables are involved, use a fully qualified column name.

The list of allowed tables includes the table aliases.

SQL functions in place of column names: Values in this tag are treated as column names if the type is set to TABLE. The column name is used directly while constructing a query, so an SQL function can be used on a column.

- Example 1

```
<Source type="TABLE">GENDER_FLAG</Source>
```

- Example 2

```
<Source type="TABLE">DECODE (GENDER_FLAG, 'F', 1, 2)</Source>
```

Attributes:

1. type:
 - Supported values = {TABLE, CONSTANT}
 - TABLE specifies that the data comes from a database table.
 - CONSTANT specifies that the data is given in the value column of this tag.

2. table: Use this tag only if the intended column isn't from the table given in the tableName attribute of this Table tag. If this tag isn't used, the column is searched for in the table given in tableName.

- o Supported values:
 - {BEN_EXTRACT_REQ_DETAILS
 - PER_ALL_PEOPLE_F
 - PER_PERSONS
 - PER_ALL_ASSIGNMENTS_M
 - PER_PEOPLE_LEGISLATIVE_F
 - BEN_PL_F
 - BEN_PL_TYP_F
 - BEN_OPT_F
 - BEN_PGM_F

Elements (Subtags)	Parent Tag
None	Field

Width

Description: Specifies the intended width of this field in the extract file. The number passed is the number of character spaces on the file.

Attributes: Supported values are positive integers.

Elements (Subtags)	Parent Tag
None	Field

Padding

Description: Specifies the alignment of data in each column.

Attributes: Supported values: {LEFT, RIGHT}

Elements (Subtags)	Parent Tag
None	Field

Table Aliases

Allowed Table	Alias
BEN_EXTRACT_REQ_DETAILS	REQ
PER_ALL_PEOPLE_F	PEO
PER_PERSONS	PER
PER_ALL_ASSIGNMENTS_M	ASG
PER_PEOPLE_LEGISLATIVE_F	LEG
BEN_PL_F	PLN
BEN_PL_TYP_F	TYP
BEN_OPT_F	OPT
BEN_PGM_F	PGM

Sample XML Layout

```
<?xml version="1.0" encoding="utf-8"?>
<Layout>
<Table tableName="DUAL">
  <RecordType>FIXEDWIDTH</RecordType>
  <Field>
    <Name>"Record Type"</Name>
    <Source type="CONSTANT">001</Source>
    <Width>3</Width>
    <Padding>Left</Padding>
  </Field>
</Table>
<Table tableName="BEN_EXTRACT_REQ_DETAILS">
  <RecordType>CSV</RecordType>
  <Field>
    <Name>"Last Name"</Name>
    <Source type="TABLE">LAST_NAME</Source>
    <Width>25</Width>
    <Padding>Left</Padding>
  </Field>
  <Field>
    <Name>"First Name"</Name>
    <Source type="TABLE">FIRST_NAME</Source>
    <Width>50</Width>
    <Padding>Left</Padding>
  </Field>
</Table>
```

```

</Field>
<Field>
  <Name>"Filler"</Name>
  <Source type="CONSTANT">XXXXXXXXXX</Source>
  <Width>10</Width>
  <Padding>None</Padding>
</Field>
<Field>
  <Name>"Plan Name"</Name>
  <Source type="TABLE">PLAN</Source>
  <Width>70</Width>
  <Padding>Left</Padding>
</Field>
<Field>
  <Name>"Coverage Start Date"</Name>
  <Source type="TABLE">COVERAGE_START_DATE</Source>
  <Width>15</Width>
  <Padding>Left</Padding>
</Field>
<Field>
  <Name>"SSN"</Name>
  <Source type="TABLE">NATIONAL_IDENTIFIER</Source>
  <Width>12</Width>
  <Padding>Left</Padding>
</Field>
<Field>
  <Name>"Gender"</Name>
  <Source type="TABLE">DECODE (GENDER_FLAG, 'F', 1, 2)</Source>
  <Width>1</Width>
  <Padding>Left</Padding>
</Field>
<Field>
  <Name>"Person Number"</Name>
  <Source type="TABLE" table="PER_ALL_PEOPLE_F">PERSON_NUMBER</Source>
  <Width>30</Width>
  <Padding>Left</Padding>
</Field>
<Field>
  <Name>"Country of Birth"</Name>
  <Source type="TABLE" table="PER_PERSONS">COUNTRY_OF_BIRTH</Source>
  <Width>30</Width>
  <Padding>Left</Padding>
</Field>
<Field>
  <Name>"Assignment type"</Name>
  <Source type="TABLE" table="per_all_assignments_m">assignment_type</Source>
  <Width>30</Width>
  <Padding>Left</Padding>
</Field>
<Field>
  <Name>"Legislation code"</Name>
  <Source type="TABLE" table="per_people_legislative_f">LEG.LEGISLATION_CODE</Source>
  <Width>30</Width>
  <Padding>Left</Padding>
</Field>
<Field>
  <Name>"Legislation code"</Name>
  <Source type="TABLE" table="ben_pl_f">PLN.PL_ID</Source>
  <Width>30</Width>
  <Padding>Left</Padding>
</Field>
</Table>
<Table tableName="DUAL">
  <RecordType>FIXEDWIDTH</RecordType>
  <Field>
    <Name>"Record Type"</Name>

```

```
<Source type="CONSTANT">999</Source>
<Width>3</Width>
<Padding>Left</Padding>
</Field>
<Field>
<Name>"Record Type"</Name>
<Source type="SYSTEM">RECORDCOUNT</Source>
<Width>3</Width>
<Padding>Left</Padding>
</Field>
</Table>
</Layout>
```


28 Transferring Encrypted Data

Setting up Encryption for File Transfer: Procedure

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, payroll batch 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.

This topic describes how to set up encryption, decryption, and signing of files.

Encryption and Signature Keys

This table identifies 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

Importing Your Public Key

Your public key (customer-key_pub) is used for encryption of outbound files. You can decrypt the files using your private key (customer-key_priv). If you also want outbound files to be signed, then the HCM Cloud private key (fusion-key_priv) is used 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 **customer-key**.

 **Note:** You must enter **customer-key** in this field. Otherwise, the encryption APIs can't use this key for encrypting outbound files.

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.

Generating the 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.

Encrypting and Uploading Files Automatically: Procedure

Encrypt files of data with PGP encryption and transfer them automatically between your servers and Oracle WebCenter Content using APIs and web services. For example, you can encrypt and decrypt files that contain sensitive employee data or confidential documents. You write programs to collect the encrypted files from your file server. You then place them on the Oracle WebCenter Content server and call a data loader to decrypt and load the data to Oracle HCM Cloud.

 **Note:** You must set up your encryption keys before you perform these tasks.

Supported Encryption Algorithms

Oracle HCM Cloud supports the following encryption algorithms. You must ensure that you use only supported encryption algorithms.

- Cipher: AES-128, Blowfish, CAST5, 3DES

 **Note:** These cipher algorithms aren't supported: Twofish, IDEA, AES-192, and AES-256

- Compression: bzip2, zlib, zip, uncompressed
- Hash: SHA-1, SHA-256, SHA-224, SHA-512, MD5, SHA-384, RIPEMD-160

Encrypting Files

This section provides the commands to encrypt files in Microsoft Windows and Linux environments using the GnuPG encryption tool. For other tools and platforms, work with your suppliers to find the necessary commands for setting the cipher algorithm.

- 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. Perform one of these steps.
 - To encrypt a file without signing, use this command:

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

- 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.

Loading Encrypted Files

Perform the following steps to load encrypted files to Oracle HCM Cloud from the Oracle WebCenter Content server.

1. Write programs to send your encrypted files to Oracle WebCenter Content, using the Oracle WebCenter Content Web Services. If your home page is: <https://Hostname/homePage/faces/AtkHomePageWelcome>, then the Oracle WebCenterContent Server WSDL is: <https://Hostname/idcws/GenericSoap?wsdl>.
2. Call the loader program to pass the encryption parameter with other required parameters. The loaderIntegrationService uses the submitEncryptedBatch method, which has an additional parameter named encryptType. This parameter has the following values, which are defined in the ORA_HRC_FILE_ENCRYPT_TYPE lookup type:
 - NONE
 - PGPSIGNED
 - PGPUNSIGNED

Transferring Files Automatically from HCM Extracts with PGP Encryption: Procedure

Transfer encrypted files to Oracle WebCenter Content using HCM Extracts and your encryption key. HCM Extracts can generate encrypted output and store it on the WebCenter Content server. For example, you can encrypt and decrypt files that contain sensitive employee data or confidential documents. Use HCM Extracts to generate encrypted files and deliver them to the WebCenter Content server. You write your own programs to collect the files.

 **Note:** You must set up your encryption keys before you try to encrypt or decrypt data.

Outbound Integrations

Set up the following information to use HCM Extracts with your outbound integrations:

1. In the Data Exchange work area, select the Manage Extract Definitions task.
2. Select the HCM Connect delivery type on the Deliver page.

 **Tip:** HCM Connect is a delivery type in HCM Extracts. You don't have to install or license HCM Connect to use this option for encryption.

3. Enter an Integration Name. The application uses this name to create the title of the entry in WebCenter Content.

4. Select an Encryption Mode. The encryption mode is one of the values from the `ORA_HRC_FILE_ENCRYPT_TYPE` lookup type. It determines how the application encrypts the file before loading it to WebCenter Content. When HCM Extracts transfers the file to WebCenter Content it generates a content ID automatically with the following format: UCMFAnnnnnn.

The file includes the following properties:

Field Name	Value
Author	FUSION_APPS_HCM_ESS_APPID
Security Group	FAFusionImportExport
Account	hcm/ dataloader/ export
Title	HEXTV1CON_ {Integration Name}_ {Encryption Type}_ {Date Time Stamp} For example: HEXTV1CON_ ExtractConn1_ PGPUNSIGNED_ 17-11-2014 14-16-44

5. Configure the HCM Extract delivery option to output an XML (data) file directly to WebCenter Content without formatting it in BI Publisher. You can achieve this by selecting Data as the output format, omitting a template name, and selecting the HCM Connect Delivery Type.
6. Download the encrypted files from WebCenter Content using client command-line tools or a web service call.

Decryption of Outbound Files

Using your private key, you can decrypt encrypted files that are generated from Oracle HCM Cloud. To verify signed files, you use the Oracle HCM Cloud public key. Ensure that these two keys are imported into the keystore. For both Microsoft Windows and Linux, use this command to decrypt both signed and unsigned files:

```
gpg --decrypt <inbound_file_name> --output <output_file_name>
```

Related Topics

- [HCM Extracts: Changes Only \(Document ID 1964920.1.\)](#)

29 Oracle Taleo Recruiting Cloud Service Integration

HCM and Oracle Taleo Recruiting Cloud Service Components: How They Work Together

The integration between HCM and Oracle Taleo Recruiting Cloud Service facilitates seamless flow of the recruitment process from creating a job requisition to hiring a worker.

HCM and Oracle Taleo Recruiting Cloud Service Integration

The integration between HCM and Oracle Taleo Recruiting Cloud Service involves the following stages:

1. HCM
 - Exporting work structures and worker data to Oracle Taleo Recruiting Cloud Service.
2. Oracle Taleo Recruiting Cloud Service
 - a. Creating a job requisition using the data imported from HCM
 - b. Selecting candidates for a job requisition
 - c. Hiring a candidate
 - d. Importing candidate data from Oracle Taleo Recruiting Cloud Service
3. HCM
 - a. Creating a pending worker
 - b. Converting the pending worker to an employee
 - c. Exporting changed data to Oracle Taleo Recruiting Cloud Service

Importing Data From Oracle Taleo Recruiting Cloud Service: Explained

Once a candidate is selected in Oracle Taleo Recruiting Cloud Service, you must import the candidate data to HCM to facilitate hiring the candidate as an employee or contingent worker in HCM.

Selected Candidates

When ready-to-be-hired candidates are imported from Oracle Taleo Recruiting Cloud Service you can view notifications:

- On the Selected Candidates tab of the New Workers to Process page
- In the Worklist notifications, if you are the hiring manager

Click the notification task to process the selected candidate as a pending worker. You must ensure that all information is correct before submitting the candidate for approval. Successfully processed pending workers appear on the Pending Workers tab. From that tab, you can convert them into employees.

Imported Data

The data that is imported from Oracle Taleo Recruiting Cloud Service to HCM includes the following:

- Candidate data, including:
 - Person and employment data
 - Compensation and payroll data
 - Education, certification, and previous experience details, which are captured as a part of HCM Talent
- Requisition data

Scheduled Processes

You can access the Import Oracle Taleo Recruiting Cloud Service Data process from the Scheduled Processes work area. You can run this process manually or schedule it to run at regular intervals (for example, weekly at a specified time). The likely volume and frequency of hiring persons in your enterprise determines how often you run this process.

After you run the import process, you can view the selected candidates in the New Person Overview page. The Selected Candidates page displays a list of candidates ready to be added as pending workers. Once you add a candidate as a pending worker, the candidate appears in the Pending Workers list. You can then convert the pending worker to an employee or a contingent worker. The imported person, payroll, compensation, and talent (education, certification, and previous experience) details appear by default in the Convert Pending Worker flow.

 **Note:** After you import a candidate from Oracle Taleo Recruiting Cloud Service to HCM, if you make any changes to the candidate or requisition in Oracle Taleo Recruiting Cloud Service, these changes aren't included in the subsequent import runs.

Exporting Data to Oracle Taleo Recruiting Cloud Service: Explained

Oracle Taleo Recruiting Cloud Service uses HCM work structures and worker data to create and process job requisitions. You can export this data from HCM to Oracle Taleo Recruiting Cloud Service.

HCM and Oracle Taleo Recruiting Cloud Service Mapping

The following table lists the business objects that HCM exports and the corresponding business objects that they are mapped to in Oracle Taleo Recruiting Cloud Service.

HCM	Oracle Taleo Recruiting Cloud Service
Departments	Departments

HCM	Oracle Taleo Recruiting Cloud Service
Jobs	Requisition Templates
Business Units	Organizations
Users	Users
Employees	Candidates
Locations	Locations
Positions	Large user-defined selection
Grades	Large user-defined selection
Legal Employers	Large user-defined selection
Job Families	Job Fields

Employees and Users Export

As a part of the integration, HCM exports employees and users such as Managers, Hiring Managers, and Recruiters, who have the privilege to create requisitions in Oracle Taleo Recruiting Cloud Service. From the HCM perspective, user and employee data are the same, and use the same data mapping. The employee or user creation process relies on certain default user types provisioned in Oracle Taleo Recruiting Cloud Service. The following table lists the roles exported from HCM, and the user types they are mapped to in Oracle Taleo Recruiting Cloud Service.

HCM Role	Oracle Taleo Recruiting Cloud Service User Type Code	Oracle Taleo Recruiting Cloud Service User Type Name
PER_RECRUITING_ADMINISTRATOR_JOB	01-RecruitingAdmin	Recruiting - Administrator
PER_POWER_RECRUITER_ABSTRACT	02-PowerRecruiter	Recruiting - Power Recruiter
PER_BASIC_RECRUITER_ABSTRACT	03-BasicRecruiter	Recruiting - Basic Recruiter
PER_LINE_MANAGER_ABSTRACT	04-PowerManager	Recruiting - Power Manager
PER_EMPLOYEE_ABSTRACT	10-Employee	Recruiting - Employee

Scheduled Processes

You can schedule the following process in the Scheduled Processes work area:

- Export All Oracle Taleo Recruiting Cloud Service Data

Exports all work structures and worker data from HCM to Oracle Taleo Recruiting Cloud Service. You must run this process typically only once, in the beginning, to send all HCM data to Oracle Taleo Recruiting Cloud Service to enable creation of job requisitions.

- Export Changed Oracle Taleo Recruiting Cloud Service Data

Exports any changed work structures and worker data from HCM to Oracle Taleo Recruiting Cloud Service. You can run this process manually or schedule it to run at regular intervals (for example, weekly at a specified time). The likely volume and frequency of hiring persons in your enterprise determines how often you run this process.

FAQs for Oracle Taleo Recruiting Cloud Service and HCM Integration

What are Large User Defined Selections (LUDS)?

Some business objects in HCM don't map directly to any business objects in Oracle Taleo Recruiting Cloud Service, when exporting data from HCM to Oracle Taleo Recruiting Cloud Service. These business objects (for example, legal employers, jobs, and positions) are mapped to Large User Defined Selections (LUDS) instead. LUDS are custom, user-defined fields in Oracle Taleo Recruiting Cloud Service, specifically configured for the integration with HCM. The mappings are defined in the Oracle Taleo Recruiting Cloud Service application.

30 Web Services

Overview

Use web services to integrate web-based applications with Oracle Human Capital Management (HCM) Cloud. Web services expose business objects and processes to other applications through the use of open standards-based technologies. For more information about web services, see:

- SOAP Web Services for Oracle HCM Cloud
- SOAP Web Services for Common Features in Oracle Applications Cloud.

Glossary

absence case

A grouping of related absences that result from the same cause used for reporting purposes.

absence category

A group of related absence types for reporting purposes.

action reason

Action reasons provide further explanation to actions, for example, an action of transfer could have reasons such as reorganization or career progression.

annualization factor

Multiplication factor used to convert base salary to an annualized amount

assignment

A set of information, including job, position, pay, compensation, managers, working hours, and work location, that defines a worker's or nonworker's role in a legal employer.

balance dimension

The scope of a balance value, such as the period of time over which it accumulates, and whether it relates to an assignment, terms, or a payroll relationship.

balance feed

Input value from an element, or all elements in a classification, that adds to, or subtracts from, a balance.

benefit rate

An amount or percentage that represents a participant's or employer's benefit offering contributions or distributions.

business unit

A unit of an enterprise that performs one or many business functions that can be rolled up in a management hierarchy.

calculation card

Captures values required for payroll calculations for some earnings and deductions, such as absence payments and involuntary deductions. For some countries, you can also create various types of cards to hold default values for tax reporting units or payroll statutory units.

compa-ratio

Salary as a percentage of salary range midpoint. 100 denotes salary at midpoint.

content item

An individual quality, skill, or qualification within a content type that you track in profiles.

cost hierarchy

The ordering of the levels at which a value for a payroll costing account segment is available for entry. The application builds each segment of the account number by checking levels of the hierarchy. For example, to build the suspense or default account number, the application checks the department and payroll levels. For the cost account number, it starts with the element entry level and continues to the payroll level until it finds a value. For the offset account, it checks only the element level.

date-effective object

An object with a change history. Professional users can retrieve the object as of a current, past, or future date.

descriptive flexfield

Customizable expansion space, such as fields used to capture additional descriptive information or attributes about an entity, such as a customer case. You may configure information collection and storage based on the context.

development goal

A goal that is geared toward facilitating the career growth of individuals so that they can perform better in their current job or prepare themselves for advancement.

division

A business-oriented subdivision within an enterprise. Each division is organized to deliver products and services or address different markets.

document category

A high level grouping of person documents such as visas, licenses, and medical certificates. Document subcategories provide further grouping of document categories.

document type

A categorization of person documents that provides a set of options to control what document information to retain, who can access the documents, whether the documents require approval, and whether the documents are subject to expiry. A document type exists for a combination of document category and subcategory.

effective end date

For a date-effective object, the end date of a physical record in the object's history. A physical record is available to transactions between its effective start and end dates.

effective sequence number

A number that identifies the order of changes made to a date-effective object in one day. On any day, the physical record with the highest number is the current record.

effective start date

For a date-effective object, the start date of a physical record in the object's history. A physical record is available to transactions between its effective start and end dates.

element

Component in the calculation of a person's pay. An element may represent a compensation or benefit type, such as salary, wages, stock purchase plans, pension contributions, and medical insurance.

element entry

The record controlling an employee's receipt of an element, including the period of time for which the employee receives the element and its value.

eligibility profile

A user-defined set of criteria used to determine whether a person qualifies for a benefits offering, variable rate or coverage, compensation plan, checklist task, or other object for which eligibility must be established.

extensible flexfield

Customizable expansion space used to capture multiple sets of information within a context or multiple contexts. Some extensible flexfields let you group contexts into categories.

fast formula

A simple way to write formulas using English words and basic mathematical functions. Formulas are generic expressions of calculations or comparisons that repeat with different input values.

flexfield

A flexible data field that you can customize to contain one or more segments or store additional information. Each segment has a value and a meaning.

flexfield segment

An extensible data field that represents an attribute and captures a value corresponding to a predefined, single extension column in the database. A segment appears globally or based on a context of other captured information.

flow

An occurrence of a flow pattern that you manage from a payroll work area or from the Data Exchange work area using the View Extracts task. The data security for your role determines which flows you can submit and access.

flow checklist

A sequence of automatic and manual flow tasks grouped into activities, such as extract reports and processes, or tasks related to payroll processing. Submitting a flow generates a checklist that you use to monitor the flow and manage its tasks.

flow pattern

A series of tasks performed in a predefined order, which are grouped into activities, such as extract reports and processes, or tasks that cover a phase of the payroll process. The flow pattern is used to generate a flow, which you can manage from its checklist.

flow task

A process or report, or manual task such as verifying results. A flow pattern can include more than one flow task.

FTE

Abbreviation for full-time equivalent, such as .5 for half-time work.

global name

A person's name in a format and language that can be understood throughout a multinational enterprise.

global transfer

The transfer of a person to another legal employer.

globals

Store values that are constant over a period of time. You can reference them in several formulas. Examples include the name of a rate, a specific date, or a company term.

goal plan

A collection of performance goals that are grouped by common characteristics, such as a specified time frame or a particular department that must work on them.

goal plan set

One or more performance goal plans that you assign as a group to a population set.

grade

A component of the employment model that defines the level of compensation for a worker.

grade ladder

A hierarchy used to group grades and define their sequence.

grade rate

Used to define pay values for grades in a legislative data group.

grade step

A level of increment within a grade.

HCM Data Loader data set

A set of data supplied together in either compressed data (.dat) files or a spreadsheet to be uploaded to Oracle HCM Cloud.

HR

Abbreviation for human resource.

job

A generic role that is independent of any single department or location. For example, the jobs Manager and Consultant can occur in many departments.

job family

A group of jobs having different but related functions, qualifications, and titles. For example, you may group a trust analyst and an operations analyst into the Analyst job family.

job role

A role, such as an accounts payable manager or application implementation consultant, that usually identifies and aggregates the duties or responsibilities that make up the job.

LDAP

Abbreviation for Lightweight Directory Access Protocol.

legal employer

A legal entity that employs people.

legislative data group

A means of partitioning payroll and related data. At least one legislative data group is required for each country where the enterprise operates. Each legislative data group is associated with one or more payroll statutory units.

local name

A person's name in a format and language that are readily understood by users in a single country but that may not be understood throughout a multinational enterprise.

lookup code

An option available within a lookup type, such as the lookup code BLUE within the lookup type COLORS.

lookup type

The label for a static list that has lookup codes as its values.

model profile

A collection of the work requirements and required skills and qualifications of a workforce structure, such as a job or position.

object group

User-defined set of elements or people that restrict the items you want to include in various processes and reports.

organization goal

A performance goal that specifies the overall objectives of the organization.

payroll batch loader

An integrated Microsoft Excel workbook loader that helps you enter data more easily into HCM tables. Used for entering balances, balance groups, elements, element entries, payroll definitions, assigned payrolls, bank information for personal payment methods, formula global values, and user-defined tables.

payroll relationship

Defines an association between a person and a payroll statutory unit based on payroll calculation and reporting requirements.

payroll statutory unit

A legal entity registered to report payroll tax and social insurance. A legal employer can also be a payroll statutory unit, but a payroll statutory unit can represent multiple legal employers.

performance goal

A results-oriented goal, often using specific targets, to assess the level of a worker's achievement.

person number

A person ID that is unique in the enterprise, allocated automatically or manually, and valid throughout the enterprise for all of a person's work and person-to-person relationships.

person type

A subcategory of a system person type, which the enterprise can define. Person type is specified for a person at the assignment level.

position

A specific occurrence of one job that is fixed within one department. It is also often restricted to one location. For example, the position Finance Manager is an instance of the job Manager in the Finance Department.

quartile

Salary range divided into four equal parts. Quartile 1 is the lowest.

quintile

Salary range divided into five equal parts. Quintile 1 is the lowest.

rating model

A scale used to measure the performance and proficiency of workers.

reference data set

Contains reference data that can be shared across a number of business units or other determinant types. A set supports common administration of that reference data.

role mapping

A relationship between one or more roles and one or more assignment conditions. Users with at least one assignment that matches the conditions qualify for the associated roles.

role provisioning

The automatic or manual allocation of a role to a user.

salary basis

Characterizes worker's base pay. Identifies payroll details used to pay base earnings, period of time pay is quoted, factor used to annualize base pay, components used to itemize adjustments into different reasons, and grade rate used for salary validation.

salary component

Change reasons that enable itemization of salary adjustments by entering amounts or percentages for one or more components, such as merit or cost of living adjustment.

set

Classified and grouped reference data that organizational entities share.

stage table

An intermediate table to which records are loaded from a spreadsheet or file and where some validation occurs.

succession plan

A plan that identifies candidates for a role or position or to succeed a named incumbent.

talent pool owner

A human resource specialist who can manage a talent pool.

talent review

A series of meetings where organization managers evaluate trends, assess strengths, and address areas of risk for the organization.

tax reporting unit

A legal entity that groups workers for the purpose of tax and social insurance reporting.

tree node

One of the branching points in a tree structure. It corresponds to a primary key in the view object of data.

tree version

An instance of a tree that includes life cycle elements such as start and end dates, and indicates whether the tree is active. If a tree is associated with a reference data set, all tree versions belong to one set.

URL

Abbreviation for uniform resource locator.

user-defined table

Structure of rows and columns that maintains date effective lists of values. Tables store values as cells for specific row and column combinations.

work relationship

An association between a person and a legal employer, where the worker type determines whether the relationship is a nonworker, contingent worker, or employee work relationship.

XML

Abbreviation for Extensible Markup Language.

