

**Oracle® Retail Merchandise Financial Planning  
Enterprise Edition Cloud Service and Assortment  
Planning Enterprise Edition Cloud Service**

Implementation Guide

Release 16.0.2

**E93075-04**

May 2018

E93075-04

Copyright © 2018, Oracle and/or its affiliates. All rights reserved.

Primary Author: Judith Meskill

Contributing Author: Scott Coulter

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.

#### **Value-Added Reseller (VAR) Language**

##### **Oracle Retail VAR Applications**

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

- (i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.
- (ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.
- (iii) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.
- (iv) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or

condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.



---

---

# Contents

<b>Send Us Your Comments</b> .....	vii
<b>Preface</b> .....	ix
Audience .....	ix
Documentation Accessibility .....	ix
Related Documents .....	ix
Customer Support .....	ix
Improved Process for Oracle Retail Documentation Corrections .....	x
Oracle Retail Documentation on the Oracle Technology Network .....	x
Conventions .....	x
<b>1 Implementation</b>	
<b>Required Skills</b> .....	1-1
<b>Batch Framework</b> .....	1-2
Batch Processes Under the Control of the Implementer .....	1-2
Batch Processes Not Under the Control of the Implementer .....	1-2
<b>Batch Framework Service Catalog</b> .....	1-2
Batch Exec Service .....	1-2
Load Measures: measload .....	1-3
Export Measures: measexport .....	1-5
Mace Calculation Service: calc .....	1-6
Load Hierarchies: hierload .....	1-6
Export Hierarchy: hierexport .....	1-6
Wait for Trigger File: waittrigger .....	1-7
Send a Trigger File: sendtrigger .....	1-8
Extract Input Files from Archive: unpack .....	1-8
Convert Informal Positions to Formal: formalize .....	1-8
Rename Positions in a Hierarchy: renamepositions .....	1-9
Workbook Refresh by Template Name: refresh .....	1-9
Workbook Delete by Template Name: delete .....	1-10
Run Workbook Build Queue: autobuild .....	1-10
Initialize Testing Environment: initrpc .....	1-10
Execute Automated Tests: runrpc .....	1-11
<b>Automated Testing with RPAC</b> .....	1-11
<b>Domain Creation</b> .....	1-13

SFTP Upload Location.....	1-13
config .....	1-13
batch_control .....	1-13
input.....	1-13
jse.....	1-14
Bootstrap Environment .....	1-14
OAT Parameters .....	1-14
Config Name .....	1-14
Partition Dim .....	1-14
JSE Jar Files.....	1-14
Batch Group .....	1-15
Overwrite .....	1-15
Domain Build.....	1-15

## A Appendix: Exit Codes

---

---

## Send Us Your Comments

Oracle Retail Merchandise Financial Planning Enterprise Edition Cloud Service and Assortment Planning Enterprise Edition Cloud Service Implementation Guide, Release 16.0.2

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

---

---

**Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

---

---

Send your comments to us using the electronic mail address: [retail-doc\\_us@oracle.com](mailto:retail-doc_us@oracle.com)

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at <http://www.oracle.com>.





---

---

# Preface

Oracle Retail Implementation Guides provide detailed information useful for implementing and configuring the application. It helps you to understand the behind-the-scenes processing of the application.

## Audience

This document is intended for the users and administrators of Oracle Retail Predictive Application Server. This may include merchandisers, buyers, and business analysts.

## Documentation Accessibility

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>.

### Access to Oracle Support

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.

## Related Documents

For more information, see the following documents in the Oracle Retail Predictive Application Server Enterprise Edition Cloud Service documentation set:

- *Oracle Retail Merchandise Financial Planning Enterprise Edition Cloud Service and Assortment Planning Enterprise Edition Cloud Service Administration Guide*
- *Oracle Retail Assortment Planning Enterprise Edition Cloud Service Release Notes*
- *Oracle Retail Merchandise Financial Planning Enterprise Edition Cloud Service Release Notes*
- *Oracle Retail Merchandise Financial Planning Enterprise Edition Cloud Service Starter Kit*
- *Oracle Retail Assortment Planning Enterprise Edition Cloud Service Starter Kit*

## Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

<https://support.oracle.com>

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

## Improved Process for Oracle Retail Documentation Corrections

To more quickly address critical corrections to Oracle Retail documentation content, Oracle Retail documentation may be republished whenever a critical correction is needed. For critical corrections, the republication of an Oracle Retail document may at times not be attached to a numbered software release; instead, the Oracle Retail document will simply be replaced on the Oracle Technology Network Web site, or, in the case of Data Models, to the applicable My Oracle Support Documentation container where they reside.

Oracle Retail documentation is available on the Oracle Technology Network at the following URL:

<http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html>

An updated version of the applicable Oracle Retail document is indicated by Oracle part number, as well as print date (month and year). An updated version uses the same part number, with a higher-numbered suffix. For example, part number E123456-02 is an updated version of a document with part number E123456-01.

If a more recent version of a document is available, that version supersedes all previous versions.

## Oracle Retail Documentation on the Oracle Technology Network

Oracle Retail product documentation is available on the following web site:

<http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html>

(Data Model documents are not available through Oracle Technology Network. You can obtain them through My Oracle Support.)

## Conventions

The following text conventions are used in this document:

Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

---

---

# Implementation

Oracle Retail Merchandise Financial Planning Enterprise Edition Cloud Service (MFP EE CS) and Oracle Retail Assortment Planning Enterprise Edition Cloud Service (AP EE CS) act as a platform to create tailored solutions or migrate existing on-premise solutions in the Merchandise Financial Planning (MFP) or Assortment Planning and Optimization (AP) solutions into the cloud. This guide addresses the process of preparing a custom solution for use in either of these Cloud Service environments.

Because Oracle Retail Cloud Service applications do not support any back-end server access, implementation for an RPAS Enterprise Edition (RPAS EE) application is different from an RPAS on-premise implementation. RPAS EE provides online tools to cover all the necessary facets of an RPAS application roll-out and administration. This includes:

- Building and patching domains from your custom configuration
- Defining nightly, weekly, or ad hoc batch process sequences
- Scheduling recurring batch processes

## Required Skills

Since RPAS EE implementations are based on a retailer- or implementer-provided configuration, working knowledge of the RPAS configuration tools is essential. The RPAS configuration tools are supported for offline use on a Windows 7 or 10 system. They are available in the RPAS EE Starter Kit, and their use is detailed in the *Oracle Retail Predictive Application Server Configuration Tools User Guide*.

In addition to supplying an RPAS configuration, the implementer must also prepare the retailer to provide RPAS hierarchy and measure data load files, as well as to take RPAS exported measure data files for any downstream integration needs. While the implementer does not call the RPAS loadHier, loadMeasure, or exportMeasure utilities directly, knowledge of their usage gained from the RPAS Administration Guide is helpful.

Data files for loading into the RPAS EE application and exported files from the RPAS EE application for integration with other systems are sent and received from the RPAS cloud environment via an SFTP site. Knowledge of the use of SFTP software, including an ability to automate such uploads and downloads, is a necessary prerequisite for routine nightly or weekly batch processing jobs.

## Batch Framework

MFP EE CS and AP EE CS operations require that the administrative user, who will not have command-line server access, must be able to select, initiate, and schedule RPAS batch activities.

The RPAS platform includes an Online Administration Tool (OAT) capability, which allows simple parameterization and scheduling of pre-configured batch tasks. The RPAS EE Cloud Service introduces an enhancement to the OAT framework that allows a sequence of several batch tasks to be defined. This sequence is built from a list of available batch services, such as measure loading, calculation, workbook refresh, and so on. These service tasks run in a defined order, so that you can know, for example, that your daily data updates have been loaded before your workbook refresh tasks are run. The RPAS EE batch tasks are configured to run under the existing OAT framework, so that scheduling them to run once, or on a repeating basis, is the same as for other OAT tasks.

The batch task sequences are defined in a small set of text files, which are specified below, with some examples.

### Batch Processes Under the Control of the Implementer

Using the RPAS EE batch execution framework, the following are under implementer control:

- List of batch operations to be run, with available parameterization
- Order in which batch operations are to be run
- Scheduling of one or more recurring batch tasks (can be modified by administrator, as needed)

### Batch Processes Not Under the Control of the Implementer

Due to the operational and security constraints of the Cloud Service environment, the following are not under implementer control:

- Parallelization: RPAS EE automatically parallelizes any applicable batch tasks with a number of processes set to match the provisioned server environment.
- Script file names, file and directory locations: Custom scripting is not supported for this environment, and no knowledge of file system names or locations is necessary in defining and parameterizing the batch task files.
- Incoming and outgoing file (ftp) location: These details are fixed within the RPAS Cloud Service environment.

## Batch Framework Service Catalog

This section describes the batch service.

### Batch Exec Service

The Batch Exec service is the controller for all the other services, specifying groups of tasks to be run, their sequences, and top-level parameters.

The Batch Exec service groups are specified in a text file `batch_exec_list.txt`. In this file, each active line takes this form:

```
batch_type | service | service parameters
```

The first column is an identifier, which may be repeated on several lines to define a grouping of tasks to be run together. The second column indicates which task from the catalogue is being requested. The third column gives parameter details for that task (as necessary). Comments may be placed in the `batch_exec_list.txt` file by starting a comment line with the hash sign (#).

Here is a sample `batch_exec_list.txt` file for reference:

```
# Daily Batch Cycle
daily | waittrigger | daily_upd.txt~ftp
daily | unpack      | daily_upd.tar.gz
daily | calc         | exp_set
daily | measexport   | daily_exp_set
daily | measload     | load_oo_list
daily | sendtrigger  | batch_load_complete.txt~ftp
daily | calc         | batch_oo

# Batch Cycle to Load OO
load_oo | measload | load_oo_list
load_oo | calc      | batch_oo

# Weekly Batch Cycle
weekly | calc      | exp_calc_set
weekly | measexport | weekly_exp_set
weekly | hierload   | clnd~14~N
weekly | hierload   | prod~14~N
weekly | hierload   | loc~14~N
weekly | measload   | load_act_list
weekly | measload   | load_oo_list
weekly | calc       | batch_week
weekly | calc       | batch_fcst
weekly | refresh    | refresh_weekly
weekly | autobuild   |
```

In this sample file, three batch task groups are specified: `daily`, `load_oo`, and `weekly`. Note that these names are implementer-defined identifiers; there is nothing special about the names "daily" or "weekly". Each identifier is thus associated with a sequence of tasks, which will run in the order they are listed in the file.

Note also that no information is provided about times or schedules on which these task groups should be run. This information is specified in the RPAS Online Administration Tool.

The services listed for each batch task group are run in the order specified when that type of batch run is requested through the OAT interface. Details on the individual batch services and what their service specs mean are detailed in the following sections.

## Load Measures: measload

The Load Measures service allows the loading of one or several measures, the data for which may be found in one or several files. The service can be configured to copy the required data files either from the incoming SFTP location or the in-cloud integration file area. The service will optionally either validate the presence of all data files and treat this as an error condition, or treat the presence of files as optional and continue with no error if the files are not present.

Groups of measures to be loaded are specified in a control file, `batch_loadmeas_list.txt`, with the columns as follows:

- Load set name
- Parameter type, which must be one of the following:
  - M - measure names (one or more of these lines may be specified)
  - V - validate option, if present indicates missing data files are to be treated as an error condition (no third-column parameter required)
  - C - copy data file source (keywords: cloud, ftp; defaults to ftp if not specified)
- Parameter value (relative to the parameter type specified above)

The file name from which each measure loads is specified in the domain configuration, so no file names are given in the batch config files. A domain configuration may register each measure in a separate file, or it may have several measures registered with the same file name.

When loading several measures in a particular batch load set, if the measures are registered with different file names, they must each be listed on their own "M" batch config line. If two or more measures are registered with the same file name, then they must be listed comma-separated on a single "M" line.

Here is an example control file for the Load Measure service:

```
# load on-order measures
load_oo|C|ftp
load_oo|V|
load_oo|M|drtyoou
load_oo|M|drtyoor,drtyooc
```

In this example, data files are to be copied from the incoming SFTP location, and if any files for the listed measures are absent, an error condition will be reported. The measure drtyoou is found in its own file, whereas the measures drtyoor and drtyooc are both to be loaded from the same file. Note that the properties for those measures will need to have been set with the same file name in the domain configuration in order for this feature to be used.

The Validate option checks for required data files in the <domain>/input directory, as well as either the FTP or Cloud data location (if the C parameter is given). This allows the measload task's Validate option to correctly detect files that were previously placed in the <domain>/input location by an unpack batch task.

When measure data files are loaded, some lines in the file may be rejected (possibly due to an incorrectly formatted input file or a position that does not exist in the dimension). The RPAS measure load process does not treat these rejected lines as errors and will continue loading any valid lines from the rest of the file. In order to detect when rejected lines were encountered (since the batch framework does not report this as an error), the loadmeas batch task writes the rejected records count into its own log file and also creates a rejected records warning file in the outgoing FTP area.

The warning file has no contents, providing all relevant information in the file name itself. The file name indicates the name of the measure, the count of rejected records, and a timestamp to indicate when the task was run.

In the example below, the measure apcpfcstslsu had four rejected records when it was loaded on 26-April-2018 at 7:52am:

```
warning.eebatch_loadmeas.apcpfcstslsu.rejected.4.20180426075212
```

## Export Measures: measexport

The Export Measures service allows the flat-file export of one or more measures, using a control file, `batch_export_list.txt`, to group (and parameterize) the measure lists for particular export operations. The control file allows multiple parameters to specify the details of each export group.

Here are the columns in the export control file:

- Export Set Name
- Parameter type, which must be one of the following:
  - M - Measure name (at least one entry required)
  - F - Filter mask measure
  - X - Base intersection (F or X is required)
  - N - Specify when to skip NA values in export (optional) (keywords: never, anyna, allna)
  - O - Output file name (optional)
  - I - Output individual files (optional)
  - S - File share destination (keywords: cloud, ftp, both)
  - C - Compress output (optional; if single file is output, compress as .gz, if multiple files, compress as .tar.gz)
  - D - Delimiter (optional character to use in place of comma; to select the | character as the delimiter, specify the keyword "PIPE")

---



---

**Note:** D simply replaces all commas with the delimiter. It does not work well with string measure values that include commas.

---



---

- Parameter value (relative to the parameter type selected above).

The optional N parameter value, if indicated, is passed as the `-skipNA` to the RPAS `exportMeasure` binary. If not given, the default value of `allna` is assumed, meaning that NA values present in the measure will *not* be written to the output. The never setting (meaning "never skip NA values") causes all NA values to be written to the output file. (Note that this may make the output file significantly larger.) The `anyana` setting causes the export to skip any row that contains an NA value.

Here is an example control file for the Export Measure service:

```
# Export PoC Plan CP
lpcp|F|lpcpexportb
lpcp|S|ftp
lpcp|M|lpcpbopc
lpcp|M|lpcpbopr
lpcp|M|lpcpbopu
lpcp|M|lpcpeopc
lpcp|M|lpcpeopr
```

For the `lpcp` export group, the implementer has given a Filter Mask measure, has indicated that the file will be published to the SFTP server location, and has given a list of several measures to be included in the output.

## Mace Calculation Service: calc

The Calc service, which indicates that mace is to be run, uses a control file called `batch_calc_list.txt`. The format of this file is as follows:

```
calc_list | [G or L] | [group or expression] | [group name or expr text]
```

The first column provides an identifier for each group of calc instructions. These identifiers are used to select calculations to be run either directly, or as part of a Batch Exec run. (The identifiers can match the Batch Exec identifier, but this is not required.) The second column specifies whether this calculation is to be run in the (G)lobal domain on across all (L)ocal domains. The third column indicates whether the calculation to be run is a registered rule group or an individual expression. The final column provides either the name of the rule group to be executed or the text of the expression to be run.

As with the other control files, any line starting with # is ignored and can be used to comment or document the file, as needed.

Here is an example file for the calculation service:

```
# Calc Set for Batch Aggregation Weekly
batch_week | G | group | Batch_GB
batch_week | L | group | Batch_AggW
batch_week | L | group | Batch_InvRoll
batch_week | G | expression | LTWPNS1sR = DRTYNS1s1R+DRTYNS1s2R
batch_week | G | expression | LTWPNS1sU = DRTYS1s1U+DRTYS1s2U-DRTYRtn1U-DRTYRtn2U

# Calc Set for Generating Forecast
batch_fcst | G | group | Batch_Fcst_G
batch_fcst | L | group | Batch_Fcst_L
```

## Load Hierarchies: hierload

The service for loading the hierarchies is slightly different from the ones previously described, as it does not require a separate control file. Each hierload instruction in the Batch Exec file provides the details for loading one hierarchy; multiple hierarchies may be loaded, each on a separate line.

The parameter column provided in the Batch Exec file contains three values, separated by the ~ character. The values are: hierarchy to be loaded, purgeAge value, and whether User Defined Dimensions (UDD) are included.

The example from the Batch Exec section, above, contains these sample values:

```
weekly | hierload | clnd~14~N
weekly | hierload | prod~14~N
weekly | hierload | loc~14~N
```

This indicates that in the weekly batch execution, the CLND hierarchy with a purgeAge value of 14 days is loaded, and there are no UDDs in the CLND hierarchy. Similar values are provided in the example for the PROD and LOC hierarchies.

## Export Hierarchy: hierexport

The Export Hierarchy service allows the flat-file export of one hierarchy using a control file, `batch_exporthier_list.txt`, to specify available options. The format of the control file is similar to the Export Measure control file, with the exception that only one hierarchy may be specified at a time.

The three columns in the control file are:



- Export Set Name
- Parameter Type, which must be one of these options:
  - H - Hierarchy Name (required)
  - T - Export Type (F - only formal, I - only informal, A [or omit] - export all positions)
  - O - Output File Name (optional, defaults to <hier>.dat)
  - C - Compress result file to .gz (optional)
  - S - File export destination (one of these three keywords: ftp, cloud, or both)
- Parameter Value (if required, by parameter type)

Here is an example control file for the Export Hier service:

```
# Export PROD hierarchy, compressed
prod_export|H|prod
prod_export|T|F
prod_export|O|prod_exp.dat
prod_export|C|
prod_export|S|ftp
```

In this example, the `prod_export` grouping indicates that only the formal positions in the PROD hierarchy will be written to a compressed file `prod_exp.dat.gz` and placed in the outgoing SFTP server location.

## Wait for Trigger File: waittrigger

For a recurring batch task (such as a nightly or weekly batch), you may schedule the batch to run at a particular time, but you also need to ensure that it will not start processing until the required input files are available. This requirement is supported by the `waittrigger` task. The trigger file is a temporary file that must be uploaded to the incoming FTP area last, after all other required files are present. Note that this trigger file will be deleted once the `waittrigger` task sees it, so you must not specify an actual data file as your trigger. For example, if the batch must wait for `prod.dat` to be present, you must specify a second file name, such as `prod_dat_trigger.txt`, and the external integration process that sends the latest `prod.dat` into the Cloud environment must also create `prod_dat_trigger.txt` after the `prod.dat` file is available.

By default, the `waittrigger` task waits for 23 hours for the trigger file to appear before timing out and reporting an error. A shorter timeout may optionally be specified, given in the number of seconds to wait.

The `waittrigger` task requires only an entry in the `batch_exec_list.txt` control file; no separate control file is required. As seen in the example above:

```
daily | waittrigger | daily_upd.txt~ftp~3600
```

This example daily batch task waits up to one hour for the file `daily_upd.txt` to be present in the incoming FTP location. The third column uses the tilde (~) character as a separator and gives two or three parameters:

- the trigger file name (simple file names only, no paths)
- a location keyword that indicates where the trigger file will be found:
  - ftp: the FTP server "input" directory
  - cloud: the RGBU Cloud data share location, used when multiple RGBU Cloud apps are integrated together

- input: the current domain's input directory
- (optional) number of seconds to wait before timing out

When the trigger file is configured to be found in the FTP area, it should be placed under the input directory (which will be the same directory location for any associated data files or data file archives).

## Send a Trigger File: **sendtrigger**

In order to notify other processes, either internal or external to the Oracle RGPU Cloud environment, of the progress of a batch task sequence, the `sendtrigger` task may be configured. This task creates a file with the specified file name. Based on the configuration of this task, the trigger file will be written to the SFTP location, the RGPU Cloud data share location, or the current domain's input directory.

The `sendtrigger` task requires only an entry in the `batch_exec_list.txt` control file; no separate control file is required. Here is an example entry for a `sendtrigger` task:

```
daily | sendtrigger | batch_load_complete.txt~ftp
```

This control line indicates that the file `batch_load_complete.txt` will be created in the SFTP area, once batch execution successfully reaches this point in the daily batch sequence.

Note that no automatic clean-up of the trigger file is performed, so other processes that look for the presence of this trigger file must remove it. If a trigger file from the previous batch run is still in place during a subsequent batch run, the file will remain in place and the file's timestamp will be updated.

## Extract Input Files from Archive: **unpack**

Batch tasks such as `hierload` or `measload` expect to find their individual `.dat` or `.ovr` files in the incoming file areas. For some integration needs, it may be preferable to send these files together in a compressed archive for faster upload and to ensure that all matching files arrive together. The `unpack` task supports this integration requirement by taking an archive file from the incoming FTP area and unpacking it into the `<domain>/input` directory. The `unpack` task will operate on archive files with the following extensions: `.tar`, `.gz`, `.tar.gz`, or `.zip`. The archive must contain only simple file names and not any subdirectory structure, as this structure would then prevent the files from being found in the `<domain>/input` directory, where later batch tasks expect them.

The daily batch example above contains this usage for the `unpack` task:

```
daily | unpack | daily_upd.tar.gz
```

The task specifies that the archive file `daily_upd.tar.gz` is expected to be in the FTP input directory, and it will be unpacked into the domain's input directory before any subsequent batch tasks are performed.

## Convert Informal Positions to Formal: **formalize**

The `formalize` service allows the modifying of current informal positions (which were created on a given hierarchy and dimension having Dynamic Position Maintenance, or DPM, enabled) to make them formal positions. One or more files matching the pattern `<hier>.formalize[.extension]` must be uploaded via SFTP (or present in the Cloud Integration directory, if exported from another Cloud Service application). This file (or files) specifies which informal positions to formalize. An option is also available to

allow the formalization of all current informal positions on a dimension. Parameters are specified via the `batch_formalizepositions_list.txt` batch control file.

The `batch_formalizepositions_list.txt` control file contains multiple lines to specify each formalize task, each with three required columns, as follows:

1. Formalization Set Name
2. Parameter Type, from these values
  - C - Column index (starting from 1) of the dimension position name in the hierarchy file (as generated by `exportHier`) [required]
  - D - Dimension to formalize [required]
  - H - Hierarchy of the dimension [required]
  - A - Formalize all informal positions on Dim (no value needed) [optional; omit if sending a `<hier>.formalize.dat` file]
3. Parameter Value (varies by parameter type)

If the (A)ll option is not provided, then at least one `<hier>.formalize[.extension]` file is expected to be available in either the incoming SFTP server or the RGPU cloud data share location (or already present in the Domain input directory, although this is unlikely in the cloud usage). If no formalize files are present, the batch task will report an error. File format details are available in the *Oracle Retail Predictive Application Server Fusion Client Administration Guide*.

Here is an example specifying a formalization task called `prod_sku`, which operates on positions of the SKU dimension in the PROD hierarchy:

```
prod_sku|H|prod|
prod_sku|D|sku|
prod_sku|C|1|
```

## Rename Positions in a Hierarchy: `renamepositions`

The Rename Positions service enables the renaming of existing positions in a domain hierarchy. The task is configured by specifying the hierarchy on which positions are to be renamed and looks for a file with the name `[hier].rn.dat` in the incoming FTP location (or the in-cloud data share location if multiple applications are integrated). If the rename data file is not present, the rename task will exit without error so that the following batch sequence steps may continue.

The format of the rename data file is as specified in the *Oracle Retail Predictive Application Server Fusion Client Administration Guide*. (See details on the RPAS `renamePositions` utility in Chapter 9, "Hierarchy Management".)

The rename positions batch task does not require a separate control file, but may be specified as an entry in the `batch_exec_list.txt` file, for example:

```
weekly | renamepositions | prod
```

This control line indicates that the weekly batch task will look for a rename positions data file for the PROD hierarchy, `prod.rn.dat`, and will carry out the renamings specified.

## Workbook Refresh by Template Name: `refresh`

The Workbook Refresh service enables the refresh of particular workbooks. This allows the selection of all workbooks built from a particular template, including the ability to match on partial template names.

The `batch_refresh_list.txt` contains only two columns: a refresh group identifier and a workbook template name pattern to match. Here is an example of this file:

```
refresh_weekly | mt_wb
refresh_weekly | mp_wb
refresh_weekly | lt_wb
refresh_weekly | lp_wb
```

The example contains only one refresh group, with four workbook template pattern names to match. All workbooks in the global domain or any local domains that are built from templates matching those patterns will be refreshed.

## Workbook Delete by Template Name: delete

The Workbook Delete service enables the bulk deletion of all workbooks built from a particular template. This service does not require a separate control file but can be fully specified within the `batch_exec_list.txt` file. All workbooks built from the given template, either at the master domain or the subdomain levels, will be removed. To remove workbooks from several template types, specify one delete task for each template.

Here is an example of an entry in the `batch_exec_list.txt` file for this task:

```
weekly | delete | AD_POC
```

## Run Workbook Build Queue: autobuild

The autobuild service is the simplest to configure in the EECS batch framework, as it requires no parameters to be specified. When the autobuild service is included in a batch task group, the `wbbatch` utility is run (on master and local domains) to invoke the `-startQueue` build option. Any workbooks that have been previously queued for automatic build will be created by this call. Since no further parameters are needed, you will note in the Batch Exec section above, that there is no third column for the autobuild service line.

## Initialize Testing Environment: initrpc

This is the first of two batch tasks that work together to provide automated test capabilities. See the full explanation of RPAC test automation capabilities in [Automated Testing with RPAC](#); the specific activities carried out by the batch tasks are described briefly here.

The `initrpc` task serves two purposes related to setting up the environment to be ready to run your automated test cases. First, it checks for new or updated test collateral files on the FTP server (there are three archives of test collateral files that can be sent: `tests.tar.gz`, `input.tar.gz`, and `compare.tar.gz`; in each case `.zip` is a supported alternative to `.tar.gz`). If any of these collateral file archives are present in the FTP area under the `rpc` subdirectory, then they will be moved into the internal holding area, ready to be used by the next step in the process; if no new files are present, then the previously sent files will continue to be used.

---

---

**Note:** Incrementally adding test collateral files are not supported; previous file sets of each type are removed before unpacking the new archive, so any updated archive must contain all collateral files of that type. (This prevents stale test scripts or data files from being left in the testing environment, which could otherwise cause unexpected test failures.)

---

---

The second task carried out by `initrpc` is to place the contents of the `input.tar.gz` (or `input.zip`) into the `<domain>/input` directory. This will be used to place any hierarchy load (`.dat`) or measure load (`.ovr`, `.clr`, `.rpl`) files into position so that subsequent batch tasks may set the domain into a known state, ready for automated tests to run and verify the expected result values.

The `initrpc` task entry in the `batch_exec_list.txt` control file does not require any parameters. It would normally be placed as the first entry in a test-enabled alternate version of a daily or weekly batch execution sequence (see full example in [Automated Testing with RPAC](#)):

```
rpc_validate | initrpc |
```

## Execute Automated Tests: `runrpc`

This is the second of the two batch tasks that work together to support automated testing capabilities. The `runrpc` task executes all automated tests in a single test .XML file (see further information about the RPAC automation testing framework in [Automated Testing with RPAC](#)). While the preceding `initrpc` task must only be run once, you may specify as many `runrpc` tasks as needed to execute all configured automation tests (possibly at several different points in an overall batch execution sequence).

The `runrpc` task entry in the `batch_exec_list.txt` control file takes one parameter that combines an identifier for the test, along with the filename of the test .XML file to be executed:

```
rpc_validate | runrpc | MFPCS_Sample_Test_1~RT01_MT_WB.xml
```

In this case, the test file `RT01_MT_WB.xml` will be executed under an identifying title "MFPCS\_Sample\_Test1". See [Automated Testing with RPAC](#) for a full example of a test-enabled batch execution sequence.

Summary test results will be visible in the output log for the batch execution (visible in the Online Admin dashboard), and full test result details will be available in the log file archive that is sent to the FTP server after the batch execution completes.

## Automated Testing with RPAC

The RPAS Pluggable Automation Component (RPAC) utility is supported for use with Enterprise Edition deployments. RPAC tests are specified in XML-format text files and cover a range of RPAS Domain and Workbook activities. (Note that RPAC does not support the testing of GUI functions and is not a performance testing tool.) In order to support the validation of a newly installed or patched environment, in the context of configured daily or weekly batch operations, RPAC for Cloud deployments is supported through new entries in the EE Batch task catalog. These tasks allow a pre-production domain to be set to a known state through a combination of hierarchy load and measure load files, and then can compare both Domain and Workbook measures to known values represented either directly in the test xml files or in data comparison files (similar to a measure data load file, but used only for comparison rather than for loading).

Three types of collateral files are involved in the RPAC testing process:

- Input data file set: a group of hierarchy (`.dat`) and measure (`.ovr`, `.clr`, or `.rpl`) data files that should be loaded into the domain before any RPAC tests are run. Uploaded to FTP site in the `rpc` directory as `input.tar.gz` or `input.zip`.

- Test file set: one or more .xml files where tests and test suites are defined using the available set of RPAC tags and attributes. Uploaded to FTP site in the rpac directory as tests.tar.gz or tests.zip.
- Comparison data file set: an optional way to efficiently validate that one or more measures currently contain an expected set of values. Uploaded to the FTP site in the rpac directory as compare.tar.gz or compare.zip.

Each of these collateral file archives, once sent through the FTP interface, will be kept internally to be used every time an RPAC-enabled batch execution sequence is run. Updates to the collateral files can be sent to the FTP site before the next call of the initrpc batch task and will be brought into the active environment at that time. Note that when any of the collateral file archives is updated, the previous contents are entirely removed from the internal storage area, so the replacement archive file must be a complete set of files of that type. (This prevents stale test scripts or data files from being left in the environment.)

The two EE Batch tasks, initrpc and runrpc, are detailed in the batch task catalog in ["Initialize Testing Environment: initrpc"](#) and ["Execute Automated Tests: runrpc"](#). The initrpc task is expected to be run once, at the start of the RPAC-enabled batch execution sequence; the runrpc task can be called multiple times, including at separate points during the batch execution sequence, if needed. Here is an example batch execution sequence that shows how an existing weekly batch specification might be augmented with RPAC tests:

```
# Standard Weekly Batch Cycle
weekly | unpack      | weekly_sales.tar.gz~ftp
weekly | hierload     | prod~14~N
weekly | hierload     | loc~14~N
weekly | measload    | load_oo_list
weekly | calc        | batch_fcst
weekly | autobuild   |

# RPAC-enhanced Batch Cycle
validate | initrpc      |
validate | hierload     | prod~14~N
validate | hierload     | loc~14~N
validate | measload    | load_oo_list
validate | runrpc      | RPAC_Domain_Tests~DomainTests.xml
validate | calc        | batch_fcst
validate | runrpc      | RPAC_Workbook_Tests~WorkbookTests.xml
```

The first section, labeled "weekly", represents a weekly batch sequence that might run at midnight every Saturday. Note that updated hierarchy and measure data files for the week are sent through FTP in an archive file named "weekly\_sales.tar.gz" using the unpack task.

The second section shows how the weekly batch sequence has been augmented with RPAC tests and named "validate". Note that the unpack task from the weekly sequence has been left out, and in its place initrpc is called to place the test data input files into the domain. If new or updated RPAC test collateral files have been placed on the FTP server, they will be brought in at this point and used.

There are two sets of RPAC tests in this sequence, specified by the runrpc task entries. The first runs immediately after the hierarchy and measure files are loaded, and validates expected values in the domain. The second test set is executed after some further calculations have been run, and builds one or more workbooks, then validates values within them as well.

When RPAC-enabled batch sequences are run, the primary log file (available through the Online Admin dashboard as well as through the FTP log archive package) will show a brief summary of test results. Full test details and log files are available in the complete log archive package from the batch exec run, available in the FTP area once the execution has completed.

For full details on the contents of an RPAC test .xml file, and all the tags and attributes that are available for specifying RPAC tests, see "Appendix B: RPAS Test Automation" in *Oracle Retail Predictive Application Server Administration Guide for the Fusion Client*. Note that the latest version of this guide specifies which RPAC features are available for Cloud deployments. Due to Cloud security constraints, some RPAC features, primarily the <SHELL> tag, have been disabled; however, inclusion of RPAC tests as a step in existing batch execution sequences should fully compensate for this restriction.

## Domain Creation

This section describes domain creation.

### SFTP Upload Location

Oracle RGPU cloud services include an SFTP site for incoming and outgoing file transfers. When you upload files via SFTP, a signal must be sent to indicate that you are finished uploading files and the files are ready to be transferred internally to be available for use by the application. This signal takes the form of a top-level subdirectory named "COMMAND", containing one file named "COMPLETE" (file contents not used; file may be empty). If the COMMAND subdirectory does not yet exist, you must create it. Then, when finished uploading files as specified below, transfer the COMPLETE file into the COMMAND subdirectory.

For the purposes of building the domain, three subdirectories in the SFTP site are used:

#### **config**

For uploading the domain configuration into the cloud environment, create an archive (either .zip or .tar.gz) containing the config directory and all of its contents. This archive file must be named as <config\_name>\_config.zip or <config\_name>\_config.tar.gz. This archive file must be placed in the config subdirectory on the SFTP server. It may be updated as often as necessary in support of domain build or patch activities.

#### **batch\_control**

The set of batch process control files, as detailed in the previous section, must be uploaded to the batch\_control subdirectory within the incoming SFTP location. These files are placed into the domain environment when the domain is built and can be updated later by running the domain patch task.

#### **input**

The initial domain creation process requires at least the .dat files for all hierarchies specified in the domain configuration. Normally, it is desirable to have an initial set of measure data load files available at domain build time as well. These data files may be placed into the input directory of the SFTP server either as individual files or in a single archive using either the .zip or .tar.gz format. If uploading data files in an archive format, the name of the archive to be unpacked will be specified as an optional parameter on the Online Admin page for building the domain. Remember not to

include any directory structure in the archive file, but only simple file names. In addition to the domain build and patch processes, batch framework tasks including measload, hierload, unpack, and waittrigger will also check for incoming files in this same input directory.

### **jse**

The Enterprise Edition includes optional support for one or more Java Special Expression (JSE) extension libraries. JSE extensions are encoded into one or more Java .jar files, and the new expressions may then be used in your domain configuration. To include your .jar file(s) in the domain environment, they must be uploaded via SFTP to a subdirectory named "jse". The .jar file(s) must then be named in the Bootstrap OAT parameters, as described below.

---

---

**Note:** Remember to send a COMPLETE file into the top-level COMMAND directory once all files for the domain build process have been uploaded as described within "[SFTP Upload Location](#)".

---

---

## **Bootstrap Environment**

A newly provisioned RPAS EE cloud environment is set up with a bootstrap configuration that allows the implementer to log into the RPAS Fusion Client and access the Online Administration Tool (OAT) interface before the domain has been built. The bootstrap OAT configuration allows only tasks required to construct a domain. Once the domain has been constructed, both the domain tasks and activities as well as the bootstrap activities will be available. This allows the domain to be re-built from scratch multiple times, should this be required.

## **OAT Parameters**

A few parameters must be specified when initiating a domain build process through OAT. The implementer must supply these values:

### **Config Name**

The name under which the configuration has been saved. For those familiar with the RPAS domain construction process, this is the name that is internally passed as the -cn parameter to rpasInstall.

### **Partition Dim**

The dimension on which the domain will be partitioned. The domain is constructed with one subdomain for each position in the given dimension. This must be a level of separation that fits with the intended workflow for individual users so that, when possible, most users' daily tasks relate to only one subdomain. (This lessens contention when many users are active in the system.)

### **JSE Jar Files**

If any Java Special Expression .jar files are used by the domain configuration, they must be indicated here, by .jar file name only. If including multiple .jar files, provide a comma-separated list of .jar file names. This field is optional and must be left blank if JSE extensions are not used. If listed, the .jar files named here must be present in the incoming FTP area under the jse directory, as indicated in the previous section, and the domain build process will halt with an error if the named .jar file(s) are not present.



### Batch Group

Once a domain has been built successfully, a named group of batch operations may be specified (typically including measure data loads and mace calculations). This operation sequence must be one `batch_type` entry in the Batch Exec control file, `batch_exec_list.txt` (described above in Batch Exec service section).

### Overwrite

In the case where the domain has already been built once, and the implementer must rebuild the domain from scratch (perhaps because a non-patchable change has been made to the configuration), this option must be selected. If it is left in the default unselected state, then the domain build process will halt and report an error, rather than overwrite the existing domain.

## Domain Build

The domain build process automatically carries out the following steps:

1. Basic validation of the given config name and partition dimension.
2. Ensure that a configuration with the given config name has been uploaded.
3. If the overwrite flag is false, ensure that there is no existing domain (reports error if domain exists).
4. If the overwrite flag is true, remove the existing domain.
5. Build the domain using the config name and the partition dimension as specified in the OAT parameter screen.
6. Copy any specified Java Special Expression .jar files into the domain environment (optional).
7. Copy any users and user groups from the bootstrap domain environment into the domain environment.
8. Copy the uploaded batch control text files into the domain (from the SFTP location).
9. Run post-domain-build batch group.
10. Add the domain details into the provisioned Fusion Client configuration.

Once all these steps have completed successfully, which you may validate using the Online Administration Tool dashboard, then you may run the secondary OAT task that performs a restart on the Fusion Client. (This is required after the first time the domain is built, to enable logging in to the new domain.)



# A

---

---

## Appendix: Exit Codes

This appendix describes all non-success exit codes from the Batch Framework services and batch administration tasks.

All EE batch scripts have consistent exit codes. Codes from 1 to 22 come from the BSA framework (although only 6 and 13 are commonly used by EE batch and so are included in the table below). Codes of 30 and above are from EE batch scripts themselves and are also listed in [Table A-1](#).

[Table A-1](#) lists the common (non-success) exit codes from the EE batch scripts and the BSA framework.

**Table A-1 Common Exit Codes**

Code	Reason
6	too few args / missing arg
13	invalid domain path
30	required environment variable is not set
31	batch config file is not found
32	selected batch config entry is not found in file
33	invalid or missing info in batch config file
34	unknown error detected in RPAS utility log output
35	file/directory not found when moving or copying files
36	file/directory permission error when moving or copying files

Note that in a "live" OCI-provisioned environment, it is not expected that customers will see any of these error codes except 31 through 33. These codes indicate issues in the customer-provided batch config files.

[Table A-2](#) lists additional exit codes from `eebatch_exporthier.ksh`, `eebatch_exportmeas.ksh`, `eebatch_loadhier.ksh`, and `eebatch_loadmeas.ksh`, that result from the exit codes of the underlying RPAS binary utilities (`exportHier`, `exportMeasure`, `loadHier` and `loadMeasure`). The exit codes from the binary utilities are reported by the EE Batch Framework as being 100 more than the raw utility results. This prevents overlap between the BSA/EE script result codes and the RPAS binary utility result codes. If `loadHier` itself returns an error code of 5, then the EE batch framework will report the error as code 105.

**Table A-2 Additional Exit Codes**

<b>Script</b>	<b>Code</b>	<b>Reason</b>
eebatch_exporthier.ksh	103	Cannot run exportHier on a subdomain
eebatch_exportmeas.ksh	102	Cannot export HSA measure; generic exportMeasure error
	103	exportMeasure encountered RPAS exception, C++ exception or unknown exceptions
eebatch_loadhier.ksh	101	loadHier purgeAll got Exception
	102	calendar prepending error
	103	RPAS HierarchyException
	104	PartitionException: loadHier cannot add new position to partition dimension
	105	Cannot run loadHier while RDM repartitioning is in progress; Internal error: domain is neither Master nor Simple; other unknown exceptions
	106	Unknown exceptions
	107	Cannot run loadHier during RDM update
eebatch_loadmeas.ksh	98	Internal aggregation error
	99	Internal aggregation error; cannot load in CLR mode because the measure does not have a clear intersection; unknown internal error
	101	RPAS exception or internal error
	102	C++ exception

It is not expected that customers will encounter any of the RPAS exceptions, internal errors, or C++ exceptions, which indicate corrupted data or a programming error.