



**ORACLE® HYPERION PROFITABILITY AND COST MANAGEMENT**

*Release 11.1.2.4*

External Automation Processes Guide

**ORACLE®**

CONTENTS IN BRIEF

Introduction to Web Services .....	2
Prerequisites .....	2
The WSDL File .....	2
Using the Web Service API Reference - ProfitabilityService .....	3
Profitability Web Service Operations .....	6
Working with Custom Scripts .....	28
Using the Profitability and Cost Management Sample Client File .....	30

# Introduction to Web Services

To facilitate the running of lengthy or repetitive processes in Oracle Hyperion Profitability and Cost Management, you can create custom scripts for your organization using Oracle Web Services Manager (OWSM) to automatically invoke processes in the production environment, such as deploying Oracle Essbase cubes or transferring data, without requiring the process to be initiated by on-site personnel.

You can generate the custom scripts using a Java application programming interface (API) to invoke the web services operations for Profitability and Cost Management. For a complete list of available operations, see “[Profitability Web Service Operations](#)” on page 6.

To assist you in creating your custom scripts, a Sample Client is also included with the installation. The sample client provides the Web Services commands that are available for Profitability and Cost Management, and helps to identify data within the model.

## Prerequisites

Before you can use Web Services to create automated scripts, you must install and configure the following components as outlined in the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*:

- Run the Repository Creation Utility (RCU). See “Postconfiguration Tasks” for Profitability and Cost Management.
- Configure Oracle Web Services Manager (OWSM). See “Configuring Oracle Web Services Manager” for Profitability and Cost Management.
- Enable security for Web Services. See “Postconfiguration Tasks” for Profitability and Cost Management.

This document assumes you have a working knowledge of the following components:

- Oracle Web Services Manager
- WSDL
- XML
- XML Schema (XSD)
- SOAP

## The WSDL File

The interface for the Web Service API is defined by its Web Services Description Language (WSDL) document. WSDL is an XML-based language that describes a Web service and specifies the location of the service and the operations that the service exposes.

To view the WSDL document for Profitability and Cost Management Web Services, see <http://<localhost>:19000/profitability/ProfitabilityService?WSDL>.

## Using the Web Service API Reference - ProfitabilityService

The *Oracle Hyperion Profitability and Cost Management Web Service API Reference - Profitability Services* provides a list of the WSDL Web Services commands used in the Profitability Service Sample Client files.

For each operation, the parameters are highlighted. To navigate through the Sample File to additional information:

- Click the plus sign to expand an item and view the associated code.
- Click any link to connect to the associated explanation. For example, click `String:applicationName` to link to Type: *String*, and view available values for that type.

For specific coding details, refer to the *Oracle Hyperion Profitability and Cost Management Web Service API Reference - Profitability Services* (Web Service JavaDoc) in the OTN Documentation Library, as described in the following procedure.

➤ To access the *Web Service API Reference - Profitability Services* document:

- 1 Open the OTN Documentation Library at <http://www.oracle.com/technetwork/middleware/performance-management/documentation/index.html>.
- 2 Locate the current release.
- 3 On the side menu, select **Financial PM Applications**.
- 4 Under **Oracle Hyperion Profitability and Cost Management, Fusion Edition**, select Web Service API Reference to display the *Oracle Hyperion Profitability and Cost Management Web Service API Reference - Profitability Services* (similar to [Figure 1 on page 4](#)).

Figure 1 Title and Content for a Version of the Web Service API Reference - ProfitabilityService

**Web Service API Reference - ProfitabilityService**  
Release 11.1.2.2.00

---

Services

---

*Service: ProfitabilityService*  
Namespace: <http://profitability.webservices.epm.oracle>  
Location: [ProfitabilityService.wsdl](#)

[+Service Definition](#)

---

Interfaces (Port Types)

---

*PortType: ProfitabilityService*  
Namespace: <http://profitability.webservices.epm.oracle>  
Location: [ProfitabilityService.wsdl](#)

ProfitabilityService: Standard interface defining the methods and their definitions which Hyperion Profitability and Cost Management product will process as WebService methods.

---

**Operation: applyBulkEdit**  
Namespace: <http://profitability.webservices.epm.oracle>  
Location: [ProfitabilityService.wsdl](#)

**Input Parameters:**

[string](#) applicationName:  
[bulkEditOptionsDTO](#) bulkEditOptions:

**Output Parameters:**

[string](#) return:

[+Operation Definition](#)

## Input and Output Parameters

### Subtopics

- [Parameter Types](#)
- [Input Parameters](#)
- [Output Parameters](#)

Operations are the methods or tasks that are available through Web Services for Profitability and Cost Management. Each operation must define an input parameter and perhaps an output parameter to control the data being used, and the process or task being performed. For a complete list of available operations, see [“Profitability Web Service Operations” on page 6](#).

### Parameter Types

Input and Output parameters or other options are presented as one of the following types:

- Data Transfer Objects (DTO) - DTO is the formatting that is used to transfer or retrieve data from the Profitability and Cost Management application server. There is no behavior associated with the DTO.

For example, when you expand `bulkEditOptionsDTO`, the option specifies the object declaration and all its accessible member name details. Only field names are displayed, and no data values are populated automatically. Java developers must use the “set” and “get” methods available for each field to set a value or read its existing value.

- Boolean - Boolean responses are always TRUE or FALSE.
- String - The String type is used to specify or return String content. For example:
  - `applicationName` (The name of a registered Profitability and Cost Management application. For example, BksSP82 and BksDP30.)
  - `xxxxxStageList` - List of Stage names in the model. Example usages of this variable can be found in [Table 6, “CalcScriptOptionsDTO”](#) class. For example, `generateStageList`, `calculateStageList`.
  - `importConfigName` - Name of the import configuration created in a Profitability and Cost Management application.
  - `paths` - Valid Genealogy execution paths created in a Profitability and Cost Management application.

**Note:** In `processCalcScriptOptions`, for `clearAllStageList` and `clearCalculatedStageList`, all stages must be entered because automatic selection of subsequent stages is not available in Web Services.

## Input Parameters

Input parameters provide the information necessary to perform an operation.

There are two parameter types:

- Simple Parameter Type - These parameters that are defined using basic Java datatypes, such as “String” values. For example, if you want to fetch a list of all available POVs in an application, use the `getPOVs()` function passing the input parameter `String applicationName` - Name of the application - which is a basic Java datatype.
- Composite Parameter Type - These parameters are defined as DTOs. For example, `CopyPOVDTO`, `CubeDeployOptions`, and so on.

See [Table 4, “ClearPOVDTO”](#) and [Table 5, “CubeDeployOptionsDTO”](#).

## Output Parameters

Output parameters provide the information requested by the operation.

For some operations, such as `deleteApplication()`, no output parameter is generated.

There are two parameter types:

- Simple Parameter Type - These parameters that are defined using basic Java datatypes, such as “String” values. For example, the `TaskflowID` generated, when requesting an Essbase cube deployment or copy of POV data.

- Composite Parameter Type - These parameters are defined using composite data types. For example, `List<ApplicationDTO>` returned when requesting a list of all existing applications using `getApplications()` Web Service method.

## Profitability Web Service Operations

### Subtopics

- [Web Service Operations by Type of Application](#)
- [applyBulkEdit](#)
- [clearASOCube](#)
- [clearPOVData](#)
- [copyPOVData](#)
- [deleteApplication](#)
- [deletePOV](#)
- [deployCube](#)
- [getApplicationType](#)
- [getApplications](#)
- [getApplicationsByType](#)
- [getAssignmentRuleDefinitions](#)
- [getDriverDefinitions](#)
- [getPOVs](#)
- [getStages](#)
- [getTaskflowStatusByProcessName](#)
- [prepareDetailedViewsForReporting](#)
- [processCalcScripts](#)
- [processDetailedCalculations](#)
- [processGenealogyExecutionPaths](#)
- [processGenealogyPathsWithOutASOCubeClear](#)
- [processLedgerClearPOV](#)
- [processLedgerCopyPOV](#)
- [processLedgerDeployCube](#)
- [processMultiPOVCalcScript](#)
- [processRunLedgerCalculation](#)
- [runImportFromStaging](#)

When you set the `PortType` in the custom script to `ProfitabilityServices`, a list of pre-defined Profitability and Cost Management operations becomes available. The complete list of operations is available in the *Oracle Hyperion Profitability and Cost Management Web Service API Reference - Profitability Services* and the lists later in this section.

By invoking the required operation in the custom `ProfitabilityServicePortTypeClient` program, you can perform specific tasks, such as `getApplications` to view a list of all existing Profitability and Cost Management applications.

---

**Caution!** Ensure that any data provided as an input parameter for a Web Services request (such as the application name, stage name, POV name or other relevant data) exists in the Profitability and Cost Management database; otherwise, the operation will fail.

---

For detailed API commands for all operations, see the *Oracle Hyperion Profitability and Cost Management Web Service API Reference - Profitability Services*. This document is available from the OTN Documentation Library, as described in [“Using the Web Service API Reference - ProfitabilityService” on page 3](#)

**Note:** All Examples in the following tables are from the sample applications that are shipped as part of %EPM\_ORACLE\_HOME%\products\Profitability\samples - BksDP30 and BksSP82.

## Web Service Operations by Type of Application

The following table lists all of the Profitability and Cost Management Web Service operations with the application types to which they apply.

**Table 1** Web Service Operations with Application Types

Operation	Standard Profitability	Detailed Profitability	Management Ledger Profitability
<a href="#">applyBulkEdit</a>		✓	
<a href="#">clearASOCube</a>	✓		
<a href="#">clearPOVData</a>	✓	✓	
<a href="#">copyPOVData</a>	✓	✓	
<a href="#">deleteApplication</a>	✓	✓	✓
<a href="#">deletePOV</a>	✓	✓	✓
<a href="#">deployCube</a>	✓		
<a href="#">getApplicationType</a>	✓	✓	✓
<a href="#">getApplications</a>	✓	✓	✓
<a href="#">getApplicationsByType</a>	✓	✓	✓
<a href="#">getAssignmentRuleDefinitions</a>		✓	
<a href="#">getDriverDefinitions</a>		✓	
<a href="#">getPOVs</a>	✓	✓	✓
<a href="#">getStages</a>	✓	✓	
<a href="#">getTaskflowStatusByProcessName</a>	✓	✓	✓

Operation	Standard Profitability	Detailed Profitability	Management Ledger Profitability
<a href="#">prepareDetailedViewsForReporting</a>		✓	
<a href="#">processCalcScripts</a>	✓		
<a href="#">processDetailedCalculations</a>		✓	
<a href="#">processGenealogyExecutionPaths</a>	✓		
<a href="#">processGenealogyPathsWithoutASOCubeClear</a>	✓		
<a href="#">processLedgerClearPOV</a>			✓
<a href="#">processLedgerCopyPOV</a>			✓
<a href="#">processLedgerDeployCube</a>			✓
<a href="#">processMultiPOVCalcScript</a>			✓
<a href="#">processRunLedgerCalculation</a>			✓
<a href="#">runImportFromStaging</a>	✓	✓	

## applyBulkEdit

Use this operation to perform Bulk Edit for the given source assignment rules with destination rules, or Drivers for a Profitability and Cost Management Detailed application. Use with Detailed Profitability applications.

### Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management application to which this Bulk Edit operation is to be applied.
- Optional: *BulkEditOptionsDTO bulkEditOptions* - DTO containing information required to perform the Apply Bulk Edit Operation.

**Table 2** BulkEditOptionsDTO

Variable	Description	Example
sourceRules	A List of Source Assignment Rule names being selected for this Bulk Edit Operation	Apply All Building Activities
destinationRules	A List of Destination Assignment Rule names being selected for this Bulk Edit Operation,  <b>Note:</b> This value should be passed only with <code>BulkEditOperations.ADD_ASSIGNMENT_RULES</code> and <code>BulkEditOperations.REMOVE_ASSIGNMENT_RULES</code> .	Sales Order to Invoice

Variable	Description	Example
drivers	Name of the Driver to be applied to the selected Source Assignment Rules as part of this Bulk Edit Operation.  <b>Note:</b> Only one Driver name may be provided when using the <code>BulkEditOperations.ADD_DRIVER</code> operation; however, a list of Driver names can be provided when using the <code>BulkEditOperations.REMOVE_DRIVERS</code> operation	DRV Build Product
operation	Specify the Bulk Edit operation: <ul style="list-style-type: none"> <li><input type="radio"/> ADD_DRIVERS</li> <li><input type="radio"/> REMOVE_DRIVERS</li> <li><input type="radio"/> ADD_ASSIGNMENT_RULES</li> <li><input type="radio"/> REMOVE_ASSIGNMENT_RULES</li> </ul>	ADD_DRIVERS
povGrp	Specify dimension members names for the POV for which this Bulk Edit operation is to be applied: <ul style="list-style-type: none"> <li><input type="radio"/> povDimensionMember1</li> <li><input type="radio"/> povDimensionMember2</li> <li><input type="radio"/> povDimensionMember3</li> <li><input type="radio"/> povDimensionMember4</li> <li><input type="radio"/> povState</li> </ul>	povDimensionMember1 = 2012 povDimensionMember2 = January povDimensionMember3 = Actual Draft
selectAllRulesOrDriversForDelete	Boolean flag specifying if all the destination assignment rules or drivers should be selected for delete.  Valid values are TRUE or FALSE.  <b>Note:</b> This value should be passed only with <code>BulkEditOperations.REMOVE_DRIVERS</code> and <code>BulkEditOperations.REMOVE_ASSIGNMENT_RULES</code> .	FALSE
selectEntireStageForDelete	Boolean flag specifying if the entire stage should be selected for delete.  Valid values are TRUE or FALSE.	FALSE
comment	Specify a comment for this Bulk Edit Operation.	"Bulk Edit Operation to Add Drivers"

## Output Parameters

*@return String* - CES task ID generated for this operation.

**Note:** Use [“getTaskflowStatusByProcessName”](#) on page 17 operation to get the status of this CES taskflow name (as it is displayed on the Taskflow Status Summary).

## clearASOCube

Use this operation to clear the ASO cube for a given App name, POV and Layer combination. This operation is for Standard Profitability applications.

## Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management application to use.
- *povMemberGroupDTO pov* - POV information

**Table 3** POVMemberGroupDTO

Variable	Description	Example
povDimensionMember1	POV Dimension Member Name at Position 1	2012
povDimensionMember2	POV Dimension Member Name at Position 2	January
povDimensionMember3	POV Dimension Member Name at Position 3	Actual
povDimensionMember4	POV Dimension Member Name at Position 4	Plan
povState	POV State. Valid values are Draft, Published or Archived	Draft

- *layer layerName* - Layer Name. Examples: COST, REVENUE

## Output Parameters

*@return String* - CES Task ID generated for this task.

**Note:** Use “[getTaskflowStatusByProcessName](#)” on [page 17](#) operation to get the status of this CES taskflow name (as it is displayed on the Taskflow Status Summary).

## clearPOVData

Use this operation to clear the POV data for selection stage and other details. Use with Standard Profitability and Detailed Profitability applications.

## Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management application.
- *ClearPOVDTO clearPOVData* - POV clear options.

**Table 4** ClearPOVDTO

Variable	Description	Example
clearAssignmentRuleSelections	Boolean flag specifying whether Assignment Rule Selections should be cleared. Valid values are TRUE or FALSE.	TRUE
clearCalculationRules	Boolean flag specifying whether Calculation Rules should be cleared. Valid values are TRUE or FALSE.	TRUE

Variable	Description	Example
clearCostLayer	Boolean flag specifying whether Cost Layer should be cleared. Valid values are TRUE or FALSE.	TRUE
clearDriverSelectionExceptions	Boolean flag specifying whether Driver Selection Exceptions should be cleared. Valid values are TRUE or FALSE.	TRUE
clearDriverSelectionRules	Boolean flag specifying whether Driver Selection Rules should be cleared. Valid values are TRUE or FALSE.	TRUE
clearRegularAssignments	Boolean flag specifying whether Regular Assignments should be cleared. Valid values are TRUE or FALSE.	TRUE
clearRevenueLayer	Boolean flag specifying whether Revenue Layers should be cleared. Valid values are TRUE or FALSE.	TRUE
pov	Specify dimension member names of the POV for which this Bulk Edit operation should be applied: <ul style="list-style-type: none"> <li>● povDimensionMember1</li> <li>● povDimensionMember2</li> <li>● povDimensionMember3</li> <li>● povDimensionMember4</li> <li>● povState</li> </ul>	povDimensionMember1 = 2012 povDimensionMember2 = January povDimensionMember3 = Actual  povState = Draft
povGrp	Specify dimension member names of the POV for which this Bulk Edit operation should be applied: <ul style="list-style-type: none"> <li>● povDimensionMember1</li> <li>● povDimensionMember2</li> <li>● povDimensionMember3</li> <li>● povDimensionMember4</li> <li>● povState</li> </ul>	povDimensionMember1 = 2012 povDimensionMember2 = January povDimensionMember3 = Actual  povState = Draft
stages	Specify dimension member names of the POV for which this Bulk Edit operation should be applied: <ul style="list-style-type: none"> <li>● stages 1</li> <li>● stages 2</li> <li>● stages 3</li> <li>● stages N, stages N = [displayOrder = (int), example: 1; stageName = (string)]</li> </ul>	Ledger Data

## Output Parameters

As the POV data is cleared instantly, there is no return value (or) output parameter for this operation.

## copyPOVData

Use this operation to copy Model artifacts and Data from a Source POV combination to a Destination POV combination. This operation is equivalent to functionality supported by selecting Manage Model, then POV Manager, and then Copy on the screen. Use with Standard Profitability and Detailed Profitability applications.

### Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management application for which the copyPOVData operation is to be performed.
- *CopyPOVDTO copyPOVData* - Selection details for Copy POV functionality. All the Boolean values in this DTO correspond to check boxes available on the Manage Model, then POV Manager, then Copy screen. See [Table 4, “ClearPOVDTO”](#).

### Output Parameters

@return String - CES Task ID generated for this operation.

**Note:** Use “[getTaskflowStatusByProcessName](#)” on [page 17](#) operation to get the status of this CES taskflow name (as it is displayed on the Taskflow Status Summary).

## deleteApplication

Use this operation to delete an existing Profitability and Cost Management application, and its association with Oracle Hyperion Shared Services. Use with Standard Profitability, Detailed Profitability, and Management Ledger Profitability applications.

---

**Caution!** Exercise caution when using `deleteApplication`, because this operation deletes the entire model data for all POVs, and also associated Essbase cubes.

---

**Note:** The application will still be available in the Oracle Hyperion EPM Architect Library. You must run Diagnostics in Oracle Hyperion EPM Architect and set the status back to “Not Deployed” in order to redeploy this application to Profitability and Cost Management.

### Input Parameters

*String applicationName* - Name of the application to be deleted from the Profitability and Cost Management database, and unregistered with Oracle Hyperion Shared Services.

### Output Parameters

None.

## deletePOV

Use this operation to delete an existing POV in a Profitability and Cost Management application. Use with Standard Profitability, Detailed Profitability, and Management Ledger Profitability applications.

---

**Caution!** Exercise caution when using this operation, because all model data associated with this POV will also be deleted with this operation.

---

### Input Parameters

- *String* applicationName - Name of the Profitability and Cost Management application from which the POV is to be deleted.
- *POVMemberGroupDTO* povDTO - Specify the dimension member names of the POV to which this Bulk Edit operation should be applied. See [Table 3, “POVMemberGroupDTO”](#).
- *layerName* - Valid values: COST, REVENUE

When setting up this operation, the following conditions apply:

1. At least one POV dimension member name is required for the operation.
2. Values can only be set to the variables that are defined in the application. If the POV is defined using two POV dimensions, then only specify values for povDimensionMember1 and povDimensionMember2. Leave the other values as “NULL”.
3. The povState is only populated when the getPOVs operation is used. This field is not necessary when passing POVMemberGroupDTO as a parameter in any operation. See [Table 3, “POVMemberGroupDTO”](#).

### Output Parameters

None.

## deployCube

Use this operation to deploy or redeploy the Calculation Cube or Reporting Cube for a selected Standard Profitability application.

The *CubeDeployOptionsDTO* options relate to the radiobuttons and check boxes in the application when you select Calculate, ...".

### Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management application Calculation or Reporting Cube for which is to be deployed or redeployed.
- *CubeDeployOptionsDTO* cubeDeployOptions - Enter selection details for deploying the cube.

**Table 5** CubeDeployOptionsDTO

Variable	Description	Example
cubeType	Valid values: <ul style="list-style-type: none"> <li>○ CALCULATION_CUBE</li> <li>○ REPORTING_CUBE</li> </ul>	CALCULATION_CUBE  REPORTING_CUBE
firstTimeDeployment	Boolean flag specifying whether the cube is being deployed for the first time for this application. Valid values are TRUE or FALSE.	FALSE
updateDatabase replaceDatabase	Both variables relate to the radio buttons in the application and therefore only one of them could and has to be set to TRUE.  Boolean flag specifying whether the database should be updated. Valid values are TRUE or FALSE.  <b>Note:</b> All the other parameters archiveDataBeforeDeploy, archiveDataAndReloadAfterDeploy, deleteDataArchiveAfterReload are only possible to set to TRUE when updateDatabase is set to TRUE	TRUE
archiveDataBeforeDeploy	Boolean flag specifying whether the data should be archived before deployment begins. Valid values are TRUE or FALSE.	TRUE
archiveDataAndReloadAfterDeploy	Boolean flag specifying whether the data archived before deployment should be reloaded after deployment completes. Valid values are TRUE or FALSE.	TRUE
deleteDataArchiveAfterReload	Boolean flag specifying whether to delete the archived data after reload. Valid values are TRUE or FALSE.	TRUE

## Output Parameters

*@return String*- CES taskflow ID generated for the deploy cube action.

**Note:** Use “[getTaskflowStatusByProcessName](#)” on page 17 operation to get the status of this CES taskflow name (as it is displayed on the Taskflow Status Summary).

## getApplicationType

Use this operation to list the application type for the existing Profitability and Cost Management application as Standard (for Standard Profitability), Detail (for Detailed Profitability), or Management Ledger (for Management Ledger Profitability).

## Input Parameters

*String applicationName* - Name of the Profitability and Cost Management application for which the application type is to be retrieved.

## Output Parameters

*@return String - ApplicationType* return -The following types are returned:

- GENERAL (For Standard Profitability applications)
- DETAIL (For Detailed Profitability applications)

## getApplications

Use this operation to list all existing Profitability and Cost Management applications. Use with Standard Profitability, Detailed Profitability, and Management Ledger Profitability applications.

## Input Parameters

None.

## Output Parameters

*@return List<ApplicationDTO>* - List of ApplicationDTOs containing application information.

## getApplicationsByType

Use this operation to list all Profitability and Cost Management applications of the selected type. Use with Standard Profitability, Detailed Profitability, and Management Ledger Profitability applications.

## Input Parameters

*String applicationType* - Specify the type of applications to be fetched from Profitability and Cost Management application server. These are the valid values:

- GENERAL (For Standard Profitability applications)
- DETAIL (For Detailed Profitability applications)

## Output Parameters

*@return List<ApplicationDTO>* - Returns a list of applications for the selected type.

## getAssignmentRuleDefinitions

Use this operation to retrieve all Assignment Rule Definitions (not associations), for a particular stage, for a given Detailed Profitability application.

## Input Parameters

- *String applicationName* - Name of the Detailed Profitability and Cost Management application for which the Assignment Rule Definitions are being retrieved.
- *String stageName* - Specify the stage name for which assignment rule definitions should be retrieved.

## Output Parameters

@return *List<AssignmentRuleDTO>* - List of AssignmentRuleDTOs matching the above input parameters.

## getDriverDefinitions

Use this operation to list all Driver definitions for a Detailed Profitability application.

## Input Parameters

*String applicationName* - Name of the Profitability and Cost Management Detailed application for which you want to view the Driver Definitions.

## Output Parameters

@return *List<DriverDTO>* - Returns a list of DriverDTOs.

## getPOVs

Use this operation to retrieve all POV details for a selected application. Use with Standard Profitability, Detailed Profitability, and Management Ledger Profitability applications.

## Input Parameters

*String applicationName* - Name of the Profitability and Cost Management application for which the POVs should be retrieved.

## Output Parameters

@return *List<POVMemberGroupDTO>* - List of POVMemberGroupDTOs containing POV information. See [Table 3, "POVMemberGroupDTO"](#) for a list of the associated members.

## getStages

Use this operation to retrieve all stage details for a selected Standard Profitability or Detailed Profitability application. You can find the name and display order of a stage by using this command.

### Input Parameters

*String applicationName* - Name of the Profitability and Cost Management application for which stage details should be retrieved.

### Output Parameters

@return *List<StageDTO>* - Returns a list of StageDTOs containing stage information.

## getTaskflowStatusByProcessName

Use this operation to view the current status of the job process name (CES taskflow) as it is displayed on the Taskflow Status Summary. The valid statuses are New, Active, Stopped, and Done. Use with Standard Profitability, Detailed Profitability, and Management Ledger Profitability applications.

### Input Parameters

*String processName* - CES process name for which status is to be retrieved.

### Output Parameters

@return *String*- Comma-separated values of all tasks and their statuses for the specified taskflow process name. This is the taskflow from the Taskflow Status screen in the application.

For example, if the process has two tasks created for it with the IDs 12345 and 123455, the task IDs and status are displayed as follows: 12345=Done, 123455=Active.

## prepareDetailedViewsForReporting

Use this operation to prepare views for a Detailed Profitability and Cost Management application.

### Input Parameters

- *String applicationName* - Name of the Detailed Profitability and Cost Management application for which the reporting views are to be prepared.
- *List<DimensionDTO>* dimensions- Specify the list of name and short name properties for dimension(s) to be included in generating the reporting views.

## Output Parameters

None.

## processCalcScripts

Use this operation to initiate the process and run calculation scripts for a selected Standard Profitability application. The following actions relate to the check boxes on the Manage Calculation tab of the application:

- Clear All
- Clear Calculated
- Generate
- Calculate
- Transfer data after calculation.

**Note:** For `clearAllStageList` or `clearCalculatedStageList`, list all the stage names that are to be cleared. If you do not want to clear any stages, use empty quotes. Note, though, that at least one stage must be added to the list of stages to be cleared. If you would prefer to clear no stages, add the last stage to the list, since it typically has no data at the time this operation is run.

## Input Parameters

- *String applicationName* - Name of the Standard Profitability and Cost Management application for which Calculation Scripts should be generated and executed, depending on the options selected.
- *CalcScriptOptionsDTO* options: - Selection details for processing Calculation scripts.

**Table 6** CalcScriptOptionsDTO

Variable	Description	Example
povGrp	POV information for which Calculation Script generation and execution should be performed. See <a href="#">Table 3</a> , "POVMemberGroupDTO".	povDimensionMember1=2012 povDimensionMember2=March povDimensionMember3=Actual povState=Draft
layerName	Layer name for which Calculation scripts should be generated and executed. Valid Values: <ul style="list-style-type: none"><li>○ COST</li><li>○ REVENUE</li></ul>	COST
clearCalculatedStageList	List of stage names for which calculated data need to be cleared.	Ledger Data, Activity
clearAllStageList	List of stage names for which all information must be cleared.	Ledger Data, Activity

Variable	Description	Example
generateStageList	List of stage names for which calc scripts need to be generated	Ledger Data, Activity
calculateStageList	List of stage names for which calc scripts should be executed	Ledger Data, Activity
transferData	Boolean flag specifying whether a data transfer need to be performed. Valid values are TRUE or FALSE.	FALSE

## Output Parameters

*@return String* - CES Task ID generated for this operation.

**Note:** Use “[getTaskflowStatusByProcessName](#)” on page 17 operation to get the status of this CES taskflow name (as it is displayed on the Taskflow Status Summary).

## processDetailedCalculations

Use this operation to process and run calculations for a selected Detailed Profitability application. The following actions relate to the check boxes on the Manage Calculation tab of the application:

- clearCalculated - “Processing Options” then “Clear Calculated Values”
- createContributionDetail - “Processing Options” then “Execute Calculations” then “Create Contribution Detail”
- createDetailCalculatedDriverTables - “Processing Options” then “Execute Calculations” then “Create Detailed Calculated Driver Tables”
- executeCalculations - “Processing Options” then “Execute Calculations”
- runSingleCalcRuleSequence - “Processing” Options” then “Run a single calculation rule sequence”
- abortOnError - “Processing Options” then “Preview with Limited Source Set”
- transferToContribDb - “Processing Options” then “Data Transfers” then “Contribution Database”
- transferToDstStgDb - “Processing Options” then “Data Transfers” then “Destination Stage Database”
- transferToSrcStgDb - “Processing Options” then “Data Transfers” then “Source Stage Database”

## Input Parameters

- *String applicationName* - Name of the Detailed Profitability and Cost Management application that is to be calculated
- *DetailedCalculationOptionsDTO* calc options - Selection details to run the calculation.

**Table 7 DetailedCalculationOptionsDTO**

Variable	Description	Examples
clearCalculated	Boolean flag specifying whether previously calculated values should be cleared. Valid values are TRUE or FALSE.	TRUE
executeCalculations	Boolean flag specifying whether calculations should be executed as part of this operation. Valid values are TRUE or FALSE. <b>Note:</b> When the executeCalculations flag is set to TRUE, you must provide values for createContributionDetail, createDriverTables, and runSingleCalcRuleSequence.	TRUE
createContributionDetail	Boolean flag specifying contribution detail should be created. Valid values are TRUE or FALSE.	TRUE
createDetailCalculatedDriverTables	Boolean flag specifying whether calculated driver tables should be created.	TRUE
dataPOVMemberGroups	List of POV Dimension Member Group details that should be considered for Data POV when calculating. See <a href="#">Table 3, "POVMemberGroupDTO"</a> .	povDimensionMember1=2012 povDimensionMember2=January povDimensionMember3=Actual
modelPOVMemberGroup	Model POV dimension member group details when performing calculations.	povDimensionMember1=2012 povDimensionMember2=January povDimensionMember3=Actual
runSingleCalcRuleSequence	Boolean flag specifying whether a single calculation rule sequence should be considered when calculating . Valid values are TRUE or FALSE.	FALSE
postScript	Name of the post-calculation script	POST
preScript	Name of the pre-calculation script	PRE
calcRuleSequence	Calculation Rule Sequence Value.	Valid values are 1 - 9999.
transferToSrcStgDb	Boolean flag specifying whether data transfer should happen to Source Stage Database.	Valid values are TRUE or FALSE.
transferToDstStgDb	Boolean flag specifying whether data transfer should happen to Destination Stage Database.	Valid values are TRUE or FALSE.
transferToContribDb	Boolean flag specifying whether data transfer should happen to Contribution Database.	Valid values are TRUE or FALSE.
abortOnError	Valid values are TRUE or FALSE.	TRUE
comment	(string)	optional field

## Output Parameters

@return String - CES Task ID generated for this operation.

**Note:** Use “[getTaskflowStatusByProcessName](#)” on page 17 operation to get the status of this CES taskflow name (as it is displayed on the Taskflow Status Summary).

## processGenealogyExecutionPaths

Use this operation to execute the genealogy paths that have been defined for a selected Standard Profitability application. The following actions relate to the check boxes when you select Calculate, then Manage Calculation, and then the Genealogy tab.

## Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management application for which the genealogy paths are to be calculated.
- *GenealogyOptionsDTO genealogyInfo* - Selection details for executing genealogy paths.

**Table 8** GenealogyOptionsDTO

Variable	Description	Example
layerName	Layer name for which the genealogy execution paths should be performed. The following are valid values: <ul style="list-style-type: none"><li>○ COST</li><li>○ REVENUE</li></ul>	Layer.COST
paths	List of genealogy execution paths	1-3-5
povGrp	Specify the POV dimension member group information pertaining to this genealogy paths execution. See <a href="#">Table 3, “POVMemberGroupDTO”</a> .	povDimensionMember1=2012  povDimensionMember2=January  povDimensionMember3=Actual  povState=Draft

## Output Parameters

@return String - CES Task ID generated for this operation.

**Note:** Use “[getTaskflowStatusByProcessName](#)” on page 17 operation to get the status of this CES taskflow name (as it is displayed on the Taskflow Status Summary).

## processGenealogyPathsWithoutASOCubeClear

Use this operation to execute the Genealogy contribution paths without clearing the ASO cube for that POV and layer combination. This operation is for use with Standard Profitability applications.

**Note:** The intended use of this web services operation is to run genealogy for multiple POVs in Standard Costing Profitability and Cost Management applications. `processGenealogyPathsWithoutASOCubeClear` is not intended for running multiple genealogy calculations on the same POV.

### Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management application.
- *GenealogyOptionsDTO genealogyInfo* - Selection details for executing genealogy paths. See [Table 8](#)

### Output Parameters

*@return String* - CES Task ID generated for executint genealogy paths.

**Note:** Use “[getTaskflowStatusByProcessName](#)” on [page 17](#) operation to get the status of this CES taskflow name (as it is displayed on the Taskflow Status Summary).

## processLedgerClearPOV

Use this operation to clear model artifacts and data from a POV combination for Management Ledger applications. This operation is equivalent to the functionality supported by the Point of View Manager screen’s Clear POV Data control. Use with Management Ledger Profitability applications.

### Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management Management Ledger application for which the clearPOVData operation is to be performed.
- *LedgerClearPOVOptionsDTO clearPOVData* ([Table 9 on page 23](#)) - Selection details for Clear POV functionality. All the Boolean values in this DTO correspond to check boxes available on the Point of View Manager’s Clear POV screen.

**Table 9** LedgerClearPOVOptionsDTO

Variable	Description	Example
povGrp	Specifies dimension member names of the POV to which this Clear POV operation should be applied: <ul style="list-style-type: none"> <li>● povDimensionMember1</li> <li>● povDimensionMember2</li> <li>● povDimensionMember3</li> <li>● povDimensionMember4</li> </ul>	povDimensionMember1 = 2012 povDimensionMember2 = January povDimensionMember3 = Actual
manageRule	Boolean flag specifying whether Manage Rule data should be cleared. Valid values are TRUE or FALSE.	TRUE
inputData	Boolean flag specifying whether Input data should be cleared. Valid values are TRUE or FALSE.	TRUE
adjustmentValues	Boolean flag specifying whether adjustment values should be cleared. Valid values are TRUE or FALSE.	TRUE
allocatedValues	Boolean flag specifying whether driver allocated values data should be cleared. Valid values are TRUE or FALSE.	TRUE

## Output Parameters

*@return String*- CES Task ID generated for this operation.

**Note:** You can use the `getTaskflowStatusByProcessName` operation to get the status of this CES task ID. See [“getTaskflowStatusByProcessName”](#) on page 17.

## processLedgerCopyPOV

Use this operation to copy model artifacts and data from a Source POV combination to a Destination POV combination for Management Ledger applications. This operation is equivalent to the functionality supported by the Point of View Manager screen’s Copy POV Data control. Use with Management Ledger Profitability applications.

## Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management Management Ledger application for which the copyPOVData operation is to be performed.
- *LedgerCopyPOVOptionsDTO* copyPOVData ([Table 10, “LedgerCopyPOVOptionsDTO”](#)) - Selection details for Copy POV functionality. All the Boolean values in this DTO correspond to check boxes available on the Point of View Manager’s Copy POV screen.

**Table 10** LedgerCopyPOVOptionsDTO

Variable	Description	Example
srcPovGrp	Specifies dimension member names of the POV for which this Copy POV operation should be performed: <ul style="list-style-type: none"> <li>● povDimensionMember1</li> <li>● povDimensionMember2</li> <li>● povDimensionMember3</li> <li>● povDimensionMember4</li> </ul>	<pre>povDimensionMember1 = 2012 povDimensionMember2 = January povDimensionMember3 = Actual</pre>
destPovGrp	Specifies dimension member names of the POV to which this Copy POV operation should be applied: <ul style="list-style-type: none"> <li>● povDimensionMember1</li> <li>● povDimensionMember2</li> <li>● povDimensionMember3</li> <li>● povDimensionMember4</li> </ul>	<pre>povDimensionMember1 = 2012 povDimensionMember2 = March povDimensionMember3 = Actual</pre>
manageRule	Boolean flag specifying whether Manage Rule data should be copied. Valid values are TRUE or FALSE.	TRUE
inputData	Boolean flag specifying whether Input data should be copied. Valid values are TRUE or FALSE.	TRUE
adjustmentValues	Boolean flag specifying whether adjustment values should be copied. Valid values are TRUE or FALSE.	TRUE
allocatedValues	Boolean flag specifying whether driver allocated values data should be copied. Valid values are TRUE or FALSE.	TRUE

## Output Parameters

*@return String*- CES Task ID generated for this operation.

**Note:** You can use the `getTaskFlowStatusByProcessName` operation to get the status of this CES task ID. See [“getTaskflowStatusByProcessName”](#) on page 17.

## processLedgerDeployCube

Use this operation to deploy or redeploy the calculation cube for a selected Management Ledger Profitability application.

The LedgerDeployOptionsDTO options relate to the check boxes you see in the application when you select Deploy Cube in the Manage Database screen.

## Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management Management Ledger application for which the Deploy Cube operation is to be performed.

- *LedgerDeployOptionsDTO* ledgerDeployOptions (Table 11 on page 25) - Selection details for deploying the cube.

**Table 11** LedgerDeployOptionsDTO

Variable	Description	Example
keepData	Boolean flag specifying whether the Essbase data needs to be preserved. Valid values are TRUE or FALSE.	FALSE
replaceCube	Boolean flag specifying whether the Essbase cube should be replaced or not. Valid values are TRUE or FALSE.	TRUE
runNow	Boolean flag specifying whether to run now or later. Valid values are TRUE or FALSE.	FALSE
comment	Enables you to specify a comment for this Management Ledger Deploy Cube operation.	"Ledger Deploy Operation for deployment"

## Output Parameters

*@return String*- Process job ID generated for the Deploy Cube action.

**Note:** You can use the `getTaskFlowStatusByProcessName` operation to get the status of this CES task ID. See "[getTaskflowStatusByProcessName](#)" on page 17.

## processMultiPOVCalcScript

This operation supports multi-POV calculations for Standard Profitability applications in serial and concurrent modes. Use this operation to initiate the process and run calculation scripts for a selected Standard Profitability application for a selected range of POVs.

The following actions relate to the check boxes on the Manage Calculation tab of the application:

- ClearCalculated
- Generate
- Calculate

**Note:** The operations are performed for all stages in the application. Generated calculation scripts are distinct from the ones generated for a single POV from the user interface or Single Calc web service. Generated calculation scripts apply to all members between the starting POV group (`startPovGrp`) and ending POV group (`endPovGrp`). Only one dimension can change between the starting and ending group. Users may need to issue more than one invocation if it is not possible to express all POVs using a single expression of the starting (`startPovGrp`) and ending groups (`endPovGrp`).

## Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management Standard Profitability application to calculate.
- *MultiPOVCalcScriptOptionsDTO MultiPOVCalcScriptOptions* (Table 12 on page 26) - Selection details for processing calculation scripts.

**Table 12** MultiPOVCalcScriptOptionsDTO

Variable	Description	Example
startPOVGrp	Specifies starting POV information for which calculation script generation and execution should be performed: <ul style="list-style-type: none"> <li>● povDimensionMember1</li> <li>● povDimensionMember2</li> <li>● povDimensionMember3</li> <li>● povDimensionMember4</li> </ul>	povDimensionMember1 = 2012 povDimensionMember2 = March povDimensionMember3 = Actual povState=Draft
endPOVGrp	Specifies ending POV information for which calculation script generation and execution should be performed: <ul style="list-style-type: none"> <li>● povDimensionMember1</li> <li>● povDimensionMember2</li> <li>● povDimensionMember3</li> <li>● povDimensionMember4</li> </ul>	povDimensionMember1 = 2012 povDimensionMember2 = June povDimensionMember3 = Actual povState=Draft
layerName	Layer name for which calculation scripts should be generated and executed. Valid values: COST and REVENUE	COST
clearCalculatedData	Boolean flag specifying whether the calculated data needs to be cleared. Valid values are TRUE or FALSE.	TRUE
saveScripts	Boolean flag specifying whether the generated calculation scripts should be saved. Valid values are TRUE or FALSE.	FALSE

## Output Parameters

*@return String*- Process job ID generated for the calculation action.

**Note:** You can use the `getTaskFlowStatusByProcessName` operation to get the status of this CES task ID. See ““[getTaskflowStatusByProcessName](#)” on page 17.

## Details

The `processMultiPOVCalcScripts` operation takes in a range of POVs to do the calculation for all stages in the application. It clears all the calculated data (if `clearCalculatedData` flag is set to TRUE) and then generates and executes the calculation scripts for the POV range specified. The `saveScripts` flag determines whether the generated scripts are saved at the end of execution.

When the operation runs the next time, the generated scripts are cleared if they exist.

## processRunLedgerCalculation

Use this operation to run calculations for a selected Management Ledger Profitability application.

The LedgerCalculationOptionsDTO options relate to the check boxes you see in the application when you select Run Calculation in the Manage Calculations screen.

### Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management Management Ledger application to calculate.
- *LedgerCalculationOptionsDTO ledgerCalculationOptions* ([Table 13 on page 27](#)) - Select the processing range.

**Table 13** LedgerCalculationOptionsDTO

Variable	Description	Example
clearCalculated	Boolean flag specifying whether the Essbase data needs to be cleared. Valid values are TRUE or FALSE.	FALSE
executeCalculations	Boolean flag specifying whether Oracle Essbase cube calculations need to be executed. Valid values are TRUE or FALSE.	TRUE
runNow	Boolean flag specifying whether to run now or later. Valid values are TRUE or FALSE.	FALSE
povGrp	Specifies dimension member names of the POV for which this calculation operation should be applied: <ul style="list-style-type: none"><li>● povDimensionMember1</li><li>● povDimensionMember2</li><li>● povDimensionMember3</li><li>● povDimensionMember4</li></ul>	povDimensionMember1 = 2012 povDimensionMember2 = January povDimensionMember3 = Actual
comment	Enables you to specify a comment for this Management Ledger Run Calculation operation.	"Ledger Run Calculation Operation for deployment"
subsetStart	Specifies a ruleSet ID for the start range of the calculation	12849
subsetEnd	Specifies a ruleSet ID for the end range of the calculation.	12859
ruleName	Name of the rule for which the calculation is to be run.	allocationRule
ruleSetName	Name of the ruleSet to which ruleName belongs.	allocationRuleSet
exeType	Expected values are: ALL_RULES, RULESET_SUBSET, ONE_RULE	ONE_RULE

### Output Parameters

@return String- Process job ID generated for the calculation action.

**Note:** You can use the `getTaskFlowStatusByProcessName` operation to get the status of this CES task ID. See [“getTaskflowStatusByProcessName” on page 17](#).

## runImportFromStaging

Use this operation to invoke the selected import from Staging tables into a Standard or Detailed Profitability and Cost Management application.

### Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management application into which the import configuration will import the data.
- *String importConfigName* - Name of the import configuration to be run.

### Output Parameters

*@return String* - CES Task ID generated for executing the Import Configuration.

**Note:** Use [“getTaskflowStatusByProcessName” on page 17](#) operation to get the status of this CES taskflow name (as it is displayed on the Taskflow Status Summary).

## Working with Custom Scripts

### Subtopics

- [Custom Script Requirements](#)
- [Creating Custom Scripts](#)

The *Web Service API Reference* is intended for Java developers who want to develop their custom ProfitabilityServicePortTypeClient classes.

You can build a custom script using the Profitability and Cost Management operations.

### Custom Script Requirements

You use these commands in your custom script to invoke the web services available for Profitability and Cost Management. Each custom script requires some or all of the components described in [Table 14, “Profitability and Cost Management Custom Script Requirements”](#).

**Table 14 Profitability and Cost Management Custom Script Requirements**

<b>Script Items</b>	<b>Description</b>
<b>Namespace and Location</b>	Use the same Namespace and Location for every command or operation, including Services: <ul style="list-style-type: none"> <li>● <b>Namespace</b> identifies the Profitability web services: <code>http://profitability.webservices.epm.oracle</code></li> <li>● <b>Location</b> identifies the location of the <code>ProfitabilityService.wsdl</code> document that you are referencing.</li> </ul>
<b>Services</b>	For each script, set the service to <b>ProfitabilityService</b> to enable the defined web services operations for Profitability and Cost Management.
<b>Interfaces (Port Types)</b>	Set the <b>Port Type</b> class to <b>ProfitabilityService</b> .
<b>Operations</b>	Operations are the available Web Services. For each available operation, the following information is defined in the API Reference: <ul style="list-style-type: none"> <li>● Operation</li> <li>● Input parameters</li> <li>● Output parameters, if required</li> <li>● Operation Definition</li> <li>● Binding Operation Definition</li> </ul> Expand the topics in the API Reference to view the relevant code.
<b>Input Parameters</b>	Each operation may require input parameters. For example, <i>String</i> <code>applicationName</code> requires you to enter the name of the application for which POVs should be retrieved.
<b>Output Parameters</b>	Each operation may have Output parameters.
<b>Operation Definition</b>	For each service, there are several methods or operations that may be used to perform tasks.
<b>Binding Operation Definition</b>	A binding operation defines information about the message format and protocol details for operations and messages for the specified port type.
<b>Messages</b>	Messages are fault messages that are displayed if an exception is encountered. These messages are automatic, and no coding is required in the custom script.
<b>Elements</b>	Elements are defined in the <code>.wsdl</code> file, so no coding is required in the custom script.
<b>Type</b>	Type represents the wrapper type for the specified parameters and return parameters. No coding is required in the custom script.

## Creating Custom Scripts

When generating a custom Web Services script, you must identify the service name, and select the operations that you want to invoke.

- To create a custom script for Profitability web services:
- 1 Set up the server that is to be enabled for Web Services. See the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.
- 2 When coding, select the **ProfitabilityService** service name. (See [“Using the Profitability and Cost Management Sample Client File” on page 30](#) for more information.)
- 3 Create the Web Service Client and select the operations to be performed.
  - For a list of available operations, see [“Profitability Web Service Operations” on page 6](#).
  - For a detailed description of the parameters for each operation, refer to the *Oracle Hyperion Profitability and Cost Management External Automation Processes Guide (Web Service JavaDoc)* in the OTN Documentation Library.
- 4 **Optional:** To run the program using a `.bat` or `.sh` script, edit the existing files to match the newly created Web Services client name.

## Using the Profitability and Cost Management Sample Client File

### Subtopics

- [Setting Up the Sample Client Environment](#)
- [Using the Sample Client File](#)
- [Compiling the Code](#)

The Sample Client File for Web Services displays the commands that can be used in your custom script for automating Profitability and Cost Management tasks, and identifies data within your Profitability and Cost Management model. The sample client file is intended as a guide only, to assist you in creating your custom scripts.

## Setting Up the Sample Client Environment

- To set up the sample client environment:
- 1 Open a command window.
- 2 **Optional:** If you are not running on the same machine on which Oracle Hyperion Enterprise Performance Management Workspace is installed, copy the folder `%EPM_ORACLE_HOME%/products/Profitability/samples/wsclient` to the machine on which the sample is to be accessed.  
For example, `C:\wsclient`.
- 3 Copy the following files from the source folders listed below to the `wsclient` folder when the SecurityPolicy associated with ProfitabilityService is `USERNAME_WITH_SAML_TOKEN`.  
You can provide this as a configurable setting in `hpm_ws_client.properties` file in [step 4](#):

**Table 15 Required Files for Sample Client**

Source Folder	File Name
%EPM_ORACLE_HOME%/../user_projects/domains/EPMSys <sup>te</sup> m/config/fmwconfig	jps-config.xml
%EPM_ORACLE_HOME%/../user_projects/domains/EPMSys <sup>te</sup> m/config/fmwconfig	<<Associated keystore file.XXXX.jks>>
If you are using file-based security, %EPM_ORACLE_HOME%/../user_projects/domains/EPMSys <sup>te</sup> m/config/fmwconfig	cwallet.sso

#### 4 Edit the hpm\_ws\_client.properties file to reflect your local settings:

```
#Open the HPCM WSDL URL which is to be accessed. For example:
http://localhost:19000/profitability/ProfitabilityService?WSDL (or) {DRIVE_LETTER}:/
{FILE_PATH}/FILE_NAME.wsdl hpcm.wsdl.url=http://localhost:19000/profitability/
ProfitabilityService?WSDL

#Delimiter used to separate String literals in paramters
string.delimiter=_

# Delimiter used to separate logical entities in parameters
# For example, when passing multiple POVs at a time, please use this to delimit POVs
# For example: 2009_January_Actual#2009_March_Actual when passing 2 POVs
string.logical.delimiter=#

# SecurityPolicy associated with ProfitabilityService, that needs to be used by Sample
Client.
# The sample client is programmed to work with only one of 2 values:
# a. USERNAME_TOKEN
# b. USERNAME_WITH_SAML_TOKEN
hpcm.service.security.policy=USERNAME_TOKEN

## These next two values are needed only when the security policy is
## USERNAME_TOKEN
#HSS user name for the Profitability user.
hss.username=admin

#Password for the username above.
hss.password=password123

## These next three values are needed only when the security policy is
## USERNAME_WITH_SAML_TOKEN

# Full Path of the jps-config.xml file in use.
jps.config.file=C:/wsclient/jps-config.xml

#WSS Recipient key alias name used.
wss.recipient.key.alias=adminalias

# WSS Credential Store Framework key used.
wss.csf.key=epmpcm.credentials
```

- 5 **Optional:** If you are not running on the same machine on which the Oracle Hyperion Enterprise Performance Management Workspace is installed, download and install JDeveloper 11.1.1.6.0 locally to obtain the appropriate JAVA\_HOME and MIDDLEWARE\_HOME folders.
- 6 From a command or shell window, set the following environment variables:

**Table 16** Sample Client Environment Variables

Environment Variable	Location
JAVA_HOME	Location in which Java Development Kit is available: <ul style="list-style-type: none"> <li>● For Windows, enter <code>SET JAVA_HOME=C:/Oracle/Middleware/jdk160_29</code></li> <li>● For UNIX, enter <code>export JAVA_HOME=/usr/c/Oracle/Middleware/jdk160_29</code></li> </ul>
MIDDLEWARE_HOME	Location in which Oracle Middleware home is installed. <ul style="list-style-type: none"> <li>● For Windows, enter <code>SET MIDDLEWARE_HOME=C:/Oracle/Middleware</code></li> <li>● For UNIX, enter <code>export MIDDLEWARE_HOME=/usr/c/Oracle/Middleware</code></li> </ul>

- 7 In the command window, go to `C:\wsclient`, and then enter the following command:

```
hpm_ws_client.bat -help
```

A list of all available functions is displayed.

- 8 Use the format and operations specified in the sample client file to build your custom script. See [“Using the Sample Client File” on page 32](#).

## Using the Sample Client File

The sample client file is intended as a guide only for you to build your own custom scripts to access Oracle Hyperion Profitability and Cost Management data through Web Services. The sample client files are available at `%EPM_ORACLE_HOME%/products/Profitability/samples/wsclient`. These files have been created using Batch Script (Windows OS) and Shell Script (UNIX/Linux OS).

► To use the sample client file:

- 1 In the command window, go to `C:\wsclient`. See [“Setting Up the Sample Client Environment” on page 30](#).

- 2 Enter the following command:

```
hpm_ws_client.bat - help
```

For a list of available functions, see [“Profitability Web Service Operations” on page 6](#).

- 3 Select the operation to be performed, and enter the command in the following format:

```
hpm_ws_client.bat - help <operation_name>
```

For example, to obtain the usage details of `getPovs` operation, enter the command in the following format:

```
hpm_ws_client.bat - help getPovs
```

#### 4 To use the sample client file to perform an operation, enter the command in the following format:

```
hpm_ws_client.bat <operation_name> <<parameters>>
```

- **Example 1: List All Applications**

For example, to obtain a list of all available applications, enter the command:

```
hpm_ws_client.bat getApplications
```

- **Example 2: List All POVs**

For example, to obtain a list of all POVs for a given application, enter the command:

```
hpm_ws_client.bat getPovs <<application name>>
```

- **Example 3: Get Stages**

For example, to retrieve the stages for an application, enter the command:

```
hpm_ws_client.bat getStages <<application name>>
```

## Compiling the Code

The client sample is provided in the following formats:

- As source code (in `wsclient/src/oracle/epm/webservices/profitability/client/ProfitabilityServicePortTypeClientSample.java`)
- As a compiled binary file (in `wsclient/lib/hpcmwsclient-sample.jar`)

To successfully compile the code, you must specify the location of the `common.components.home` folder.

If the source code needs to be recompiled for any reason, you can recompile using Ant. The `build.xml` file for Ant is available in the `wsclient` folder.

► To recompile the source code:

#### 1 Open a command or shell window to specify the location of the `common.components.home` folder.

This folder is defined as `MIDDLEWARE_HOME/oracle_common`, where `MIDDLEWARE_HOME` is set as follows:

- For Windows: `SET MIDDLEWARE_HOME=C:/Oracle/Middleware`
- For UNIX: `export MIDDLEWARE_HOME=/usr/c/Oracle/Middleware`

#### 2 Pass the folder location to Ant, using one of the following methods:

- As a command line parameter. For example:

```
ant -Dcommon.components.home=C:/Oracle/Middleware/oracle_common
```
- In the `build.properties` file, open the file for editing and uncomment the definition of the `common.components.home` variable. For example:

```
common.components.home=C:/Oracle/Middleware/oracle_common
```

#### 3 Recompile the source code.

## COPYRIGHT NOTICE

Profitability and Cost Management External Automation Processes Guide, 11.1.2.4

Copyright © 2012, 2015, Oracle and/or its affiliates. All rights reserved.

Updated: June 2015

Authors: EPM Information Development Team

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.