



ORACLE® HYPERION PROFITABILITY AND COST MANAGEMENT

Release 11.1.2.2

External Automation Processes Guide

ORACLE®
ENTERPRISE PERFORMANCE
MANAGEMENT SYSTEM

CONTENTS IN BRIEF

Introduction to Web Services	2
Prerequisites	2
The WSDL File	3
Using the Web Service API Reference - ProfitabilityService	3
Profitability Web Service Operations	6
Working with Custom Scripts	19
Using the Profitability and Cost Management Sample Client File	21

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.

See the following sections:

- [“Prerequisites” on page 2](#)
- [“The WSDL File” on page 3](#)
- [“Using the Web Service API Reference - ProfitabilityService” on page 3](#)
- [“Profitability Web Service Operations” on page 6](#)
- [“Working with Custom Scripts” on page 19](#)
- [“Using the Profitability and Cost Management Sample Client File” on page 21](#)

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 *Web Service API Reference - ProfitabilityService* 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 *Web Service API Reference - ProfitabilityService* (Web Service JavaDoc) in the OTN Documentation Library, as described in the following procedure.

➤ To access the *Web Service API Reference - ProfitabilityService* 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 Select **View Library**, and then **Financial PM Applications**.
- 4 Under **Oracle Hyperion Profitability and Cost Management, Fusion Edition**, select **Web Service Java Doc** to display the Web Services API Reference for both Standard and Detailed Profitability and Cost Management.

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

The 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](#).

See the following sections:

- [“Parameter Types” on page 4](#)
- [“Input Parameters” on page 5](#)
- [“Output Parameters” on page 5](#)

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. The 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, BksSP81 or BksDP20.)
 - xxxxstageList - List of Stage names in the model. Example usages of this variable can be found in CalcScriptOptionsDTO class.
 - 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 clearAllCalculatedStageList, all stages must be entered because automatic selected 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 `getPOV()` 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`, `CubeDeployOption`, and so on.

Output Parameters

Output parameters provide the information requested by the operation.

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 Oracle Essbase cube deployment or POV copy of 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.

For some operations, such as `deletePOV`, no output parameter is generated.

Profitability Web Service Operations

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 are available from Profitability and Cost Management external automation service - *ProfitabilityService*.

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 may fail.

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

See the following list of all available operations

- [“applyBulkEdit” on page 7](#)
- [“copyPOVData” on page 8](#)
- [“deleteApplication” on page 10](#)
- [“deletePOV” on page 10](#)
- [“deployCube” on page 11](#)
- [“getApplicationType” on page 12](#)
- [“getApplications” on page 12](#)
- [“getApplicationsByType” on page 13](#)
- [“getAssignmentRuleDefinitions” on page 13](#)
- [“getDriverDefinitions” on page 13](#)
- [“getPOV” on page 14](#)
- [“getStages” on page 14](#)
- [“getTaskStatusByProcessName” on page 14](#)
- [“importFromStaging” on page 15](#)
- [“prepareDetailedViewsForReporting” on page 15](#)
- [“processCalcScripts” on page 15](#)
- [“processDetailedCalculations” on page 17](#)
- [“processGenealogyExecutionPaths” on page 18](#)

Note: All Examples in the following tables are from the sample applications that are shipped as part of %EPM_ORACLE_HOME%\products\Profitability\samples - BksDP20 and BksSP81.

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.

Input Parameters

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

Table 1 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, Note: This value should be passed only with <code>BulkEditOperations.ADD_ASSIGNMENT_RULES</code> and <code>BulkEditOperations.REMOVE_ASSIGNMENT_RULES</code> .	"Sales Order to Invoice"
drivers	Name of the Driver to be applied to the selected Source Assignment Rules as part of this Bulk Edit Operation. Note: 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"> ○ <code>BulkEditOperations.ADD_DRIVERS</code> ○ <code>BulkEditOperations.REMOVE_DRIVERS</code> ○ <code>BulkEditOperations.ADD_ASSIGNMENT_RULES</code> ○ <code>BulkEditOperations.REMOVE_ASSIGNMENT_RULES</code> 	<code>BulkEditOperations.ADD_DRIVERS</code>

Variable	Description	Example
povGrp	Specify dimension members names for the POV for which this Bulk Edit operation is to be applied: <ul style="list-style-type: none"> ○ povDimensionMember1 ○ povDimensionMember2 ○ povDimensionMember3 ○ povDimensionMember4 	povDimensionMember1 = 2012 povDimensionMember2 = January povDimensionMember3 = Actual
selectAllRulesOrDriversForDelete	Boolean flag specifying if all the destination assignment rules or drivers should be selected for delete. Valid values are TRUE or FALSE.	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: You can use `getTaskStatusByProcessName` operation to get the status of this CES task ID.

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.

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.

Table 2 CopyPOVDTO

Variable	Description	Example
sourcePOV	Specify dimension member names of the POV for which this Bulk Edit operation should be applied: <ul style="list-style-type: none"> ○ povDimensionMember1 ○ povDimensionMember2 ○ povDimensionMember3 ○ povDimensionMember4 	povDimensionMember1 = 2012 povDimensionMember2 = January povDimensionMember3 = Actual
targetPOV	Specify dimension member names of the POV for which this Bulk Edit operation should be applied: <ul style="list-style-type: none"> ○ povDimensionMember1 ○ povDimensionMember2 ○ povDimensionMember3 ○ povDimensionMember4 	povDimensionMember1 = 2012 povDimensionMember2 = March povDimensionMember3 = Actual
copyCostLayerData	Boolean flag specifying whether Cost Layer data should be copied. Valid values are TRUE or FALSE.	TRUE
copyRevenueLayerData	Boolean flag specifying whether Revenue Layer data should be copied. Valid values are TRUE or FALSE.	TRUE
copyAssignments	Boolean flag specifying whether assignments data should be copied. Valid values are TRUE or FALSE.	TRUE
copyDriverAssociations	Boolean flag specifying whether driver associations data should be copied. Valid values are TRUE or FALSE.	TRUE
copyDriverValues	Boolean flag specifying whether driver data should be copied. Valid values are TRUE or FALSE.	FALSE
copyCostRevenueValues	Boolean flag specifying whether Cost/Revenue values should be copied. Valid values are TRUE or FALSE.	FALSE
copyStageObjectCalculations	Boolean flag specifying whether Stage Object Calculations should be copied for a Profitability and Cost Management Detailed Application. Valid values are TRUE or FALSE.	FALSE

Output Parameters

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

Note: You can use `getTaskStatusByProcessName` operation to get the status of this CES task ID. See [“getTaskStatusByProcessName”](#) on page 14.

deleteApplication

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

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.

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.

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

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.

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 checkboxes you see in the application when you select Calculate, then Manage Database, and then the Calculation or Reporting Database.

Input Parameters

- *String applicationName* - Name of the Profitability and Cost Management application that is to be deployed.
- *CubeDeployOptionsDTO cubeDeployOptions* - Enter selection details for deploying the cube.

Table 4 CubeDeployOptionsDTO

Variable	Description	Example
<code>cubeType</code>	Valid values: <ul style="list-style-type: none"> ○ <code>CubeType.CALCULATION_CUBE</code> ○ <code>CubeType.REPORTING_CUBE</code> 	<code>CubeType.REPORTING_CUBE</code>
<code>firstTimeDeployment</code>	Boolean flag specifying whether the cube is being deployed for the first time for this application. Valid values are TRUE or FALSE.	FALSE
<code>updateDatabase</code>	Boolean flag specifying whether the database should be updated. Valid values are TRUE or FALSE.	TRUE
<code>replaceDatabase</code>	Boolean flag specifying whether the database should be replaced. Valid values are TRUE or FALSE.	FALSE
<code>archiveDataBeforeDeploy</code>	Boolean flag specifying whether the data should be archived before deployment begins. Valid values are TRUE or FALSE.	TRUE
<code>archiveDataAndReloadAfterDeploy</code>	Boolean flag specifying whether the data archived before deployment should be reloaded after deployment completes. Valid values are TRUE or FALSE.	FALSE

Variable	Description	Example
deleteDataArchiveAfterReload	Boolean flag specifying whether to delete the archived data after reload. Valid values are TRUE or FALSE.	FALSE

Output Parameters

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

Note: You can use “[getTaskStatusByProcessName](#)” on page 14 operation to get the status of this CES task ID.

getApplicationType

Use this operation to list the application type for all existing Profitability and Cost Management applications ad General (for Standard Profitability) or Detail (for Detailed 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:
 - `ApplicationType.GENERAL` (For Standard Profitability applications)
 - `ApplicationType.DETAIL` (For Detailed Profitability applications)

getApplications

Use this operation to list all existing Profitability and Cost Management 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.

Input Parameters

- *String applicationType* - Specify the type of applications to be fetched from Profitability and Cost Management application server. These are the valid values:
 - `ApplicationType.GENERAL` (For Standard Profitability applications)
 - `ApplicationType.DETAILED` (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 Profitability and Cost Management Detailed 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.

getPOV

Use this operation to retrieve all POV details for a selected application.

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 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.

getTaskStatusByProcessName

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 status message contains Success or Failure details of the individual stages, along with the In-Progress or Completion status of the taskflow as a whole.

Input Parameters

- *String processName* - Enter the process name or taskflow ID for which the status should 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

importFromStaging

Use this operation to execute the selected import configuration into a 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 executed.

Output Parameters

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

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 Mange Calculation tab of the application:

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

Note: If you select the `clearAllStageList` or `clearAllCalculatedStageList`, list all stages that are to be cleared. If you do not want to clear any stages, use empty quotes ""

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 5 CalcScriptOptionsDTO

Variable	Description	Example
povGrp	POV information for which Calculation Script generation and execution should be performed. See Table 3 , "POVMemberGroupDTO".	povDimensionMember1=2012 povDimensionMember2=March povDimensionMember3=Actual
layerName	Layer name for which Calculation scripts should be generated and executed. Valid Values: <ul style="list-style-type: none"> ○ Layer.COST ○ Layer.REVENUE 	Layer.COST
clearCalculatedStageList	List of stage names for which generated calc scripts need to be cleared.	Ledger Data, Activity
clearAllStageList	List of stage names for which all information must be cleared.	Ledger Data, Activity
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: You can use “[getTaskStatusByProcessName](#)” on page 14 operation to get the status of this CES task ID.

processDetailedCalculations

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

- clearCalculated - “Processing Options” then “Clear Calculation Values”
- createContributionDetail - “Processing Options” then “Execute Calculations” then “Create Contribution Detail”
- createStageBalanceDetail - “Processing Options” then “Execute Calculations” then “Create Stage Balance Detail”
- createDriverTables - “Processing Options” then “Execute Calculations” then “Create Detailed Calculated Driver Tables”
- executeCalculations - “Processing Options” then “Execute Calculations”
- limitedPreview - “Processing Options” then “Preview with Limited Source Set”
- abortOnError - “Processing Options” then “Abort Task Flow if any POV fails”

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 6 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. Note: When the executeCalculations flag is set to TRUE, you must provide values for createContributionDetail, createStageBalanceDetail, createDriverTables, limitedPreview and srcAssignRuleName.	TRUE
createContributionDetail	Boolean flag specifying contribution detail should be created. Valid values are TRUE or FALSE.	TRUE
createDriverTables	Boolean flag specifying whether previously calculated values should be cleared. Valid values are TRUE or FALSE.	TRUE

Variable	Description	Examples
createStageBalanceDetail	Boolean flag specifying whether Stage balancing detail should be created during calculation processing. Valid values are TRUE or FALSE.	TRUE
dataPOVMemberGroup	List of POV Dimension Member Group details that should be considered for Data POV when calculating. See Table 3, "POVMemberGroupDTO" .	povDimensionMember1=2011 povDimensionMember2=January povDimensionMember3=Actual
modelPOVMemberGroup	Model POV dimension member group details when performing calculations.	povDimensionMember1=2011 povDimensionMember2=January povDimensionMember3=Actual
limitedPreview	Boolean flag specifying whether limited preview should be selected and a Limited Source Set 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
srcAssignRuleName	Name of the Source Assignment Rule that needs to be considered for Limited Source Set when performing calculations. Note: This value should only be specified when the limitedPreview value is set to TRUE.	Applying Rework

Output Parameters

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

Note: You can use [“getTaskStatusByProcessName” on page 14](#) operation to get the status of this CES task ID.

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 7 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"> ○ Layer.COST ○ Layer.REVENUE 	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 Table 3, "POVMemberGroupDTO" .	povDimensionMember1=2011 povDimensionMember2=January povDimensionMember3=Actual

Output Parameters

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

Note: You can use [“getTaskStatusByProcessName” on page 14](#) operation to get the status of this CES task ID.

Working with 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.

See these sections:

- [“Custom Script Requirements” on page 19](#)
- [“Creating Custom Scripts” on page 20](#)

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 8, “Profitability and Cost Management Custom Script Requirements”](#).

Table 8 Profitability and Cost Management Custom Script Requirements

Script Items	Description
Namespace and Location	Use the same Namespace and Location for every command or operation, including Services: <ul style="list-style-type: none"> ● Namespace identifies the Profitability web services: <code>http://profitability.webservices.epm.oracle</code> ● Location identifies the location of the <code>ProfitabilityService.wsdl</code> document that you are referencing.
Services	For each script, set the service to ProfitabilityService to enable the defined web services operations for Profitability and Cost Management.
Interfaces (Port Types)	Set the Port Type class to ProfitabilityService .
Operations	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"> ● Operation ● Input parameters ● Output parameters, if required ● Operation Definition ● Binding Operation Definition Expand the topics in the API Reference to view the relevant code.
Input Parameters	Each operation may require input parameters. For example, <i>String</i> applicationName requires you to enter the name of the application for which POVs should be retrieved.
Output Parameters	Each operation may have Output parameters.
Operation Definition	For each service, there are several methods or operations that may be used to perform tasks.
Binding Operation Definition	A binding operation defines information about the message format and protocol details for operations and messages for the specified port type.
Messages	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.
Elements	Elements are defined in the <code>.wsdl</code> file, so no coding is required in the custom script.
Type	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 Hyperion 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 21](#) 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 *Web Service API Reference - ProfitabilityService (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

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.

See the following procedures:

- [“Setting Up the Sample Client Environment” on page 21](#)
- [“Using the Sample Client File” on page 23](#)
- [“Compiling the Code” on page 23](#)

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:

Table 9 Required Files for Sample Client

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

4 Edit the `hpm_ws_client.properties` file to reflect your local settings:

```
# 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

# HPCM WSDL URL which is to be accessed. Eg: 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 parameters
string.delimiter=_
```

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 10 Sample Client Environment Variables

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

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 23](#).

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 21](#).

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 `wscclient` 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.

Using the Profitability and Cost Management Sample

The following show examples of variables included in an operation.

Table 11 Examples of Sample Client Operations

Operation	Example
applyBulkEdit	<code>hpm_ws_client applyBulkEdit BksDP20 2012_January_Actual ADD_DRIVERS "Adding Drivers via Bulk Edit" FALSE FALSE "Apply All Building Activities_Apply Testing" "" "DRV Build Product"</code>
copyPOVData	<code>hpm_ws_client copyPOVData BksSP81 2011_January_Actual 2012_January_Actual TRUE TRUE TRUE TRUE FALSE FALSE</code>
deleteApplication	<code>hpm_ws_client deleteApplication BksSP81</code>
deletePOV	<code>hpm_ws_client deletePOV BksSP81 2010_January_Actual</code>
deployCube	<code>hpm_ws_client deployCube BksSP81 REPORTING_CUBE TRUE FALSE FALSE FALSE FALSE FALSE FALSE</code>
getApplicationType	<code>hpm_ws_client getApplicationType BksSP81</code>
getApplications	<code>hpm_ws_client getApplications</code>
getApplicationsByType	<code>hpm_ws_client getApplicationsByType GENERAL</code>

Operation	Example
getAssignmentRuleDefinitions	hpm_ws_client getAssignmentRuleDefinitions BksDP20 "Customer Activity Cost"
getDriverDefinitions	hpm_ws_client getDriverDefinitions BksDP20
getPOV	hpm_ws_client getPOV BksSP81
getStages	hpm_ws_client getStages BksSP81
getTaskStatusByProcessName	hpm_ws_client getTaskflowStatusByProcessName BksSP81_CopyPOV_D20120315T171205_9a0
importFromStaging	hpm_ws_client runImportConfiguration BksSP81 ImportAllArtifacts
prepareDetailedViewsForReporting	hpm_ws_client prepareDetailedViewsForReporting BksDP20 Year_Yr#Accounts_Acct#Products_Prod
processCalcScripts	hpm_ws_client processCalcScriptssOptions BksSP81 2011_January_Actual_Cost "Ledger Data_Activity" "Ledger Data_Activity" "Ledger Data_Activity" "Ledger Data_Activity" FALSE
processDetailedCalculations	hpm_ws_client processDetailedCalculations BksDP20 2011_January_Actual "Running Calculations for Jan 2011" 2011_January_Actual TRUE TRUE TRUE TRUE FALSE TRUE FALSE ""PRE POST 2011_January_Actual
processGenealogyExecutionPaths	hpm_ws_client processGenealogyExecutionPathsBksSP81 2011_January_Actual COST 1-3-5

COPYRIGHT NOTICE

Profitability and Cost Management External Automation Processes Guide, 11.1.2.2

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

Authors: EPM Information Development Team

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

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, the following notice is applicable:

U.S. GOVERNMENT RIGHTS:

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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.

This software or hardware and documentation may provide access to or information on 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. 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.