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 .............................................. 21
Using the Profitability and Cost Management Sample Client File .......... 23
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:


2. Locate the current release.

3. On the side menu, select Financial PM Applications.

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.)
  - `StageList` - List of Stage names in the model. Example usages of this variable can be found in Table 5, “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 3, “ClearPOVDTO” and Table 4, “CubeDeployOptionsDTO”.

### Output Parameters

Output parameters provide the information requested by the operation.

For some operations, such as `deleteApplications()`, 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 Oracle 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

- applyBulkEdit
- clearASOCube
- clearPOVData
- copyPOVData
- deleteApplications
- deletePOV
- deployCube
- getApplicationType
- getApplications
- getApplicationsByType
- getAssignmentRuleDefinitions
- getDriverDefinitions
- getPOVs
- getStages
- getTaskflowStatusByProcessName
- prepareDetailedViewsForReporting
- processCalcScripts
- processDetailedCalculations
- processGenealogyExecutionPaths
- processGenealogyPathsWithOutASOCubeClear
- runImportFromStaging

When you set the PortType in the custom script to ProfitabilityServices, a list of predefined Profitability and Cost Management operations becomes available. The complete list of operations are available from Oracle Hyperion Profitability and Cost Management Web Service API Reference - Profitability Services.

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, 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.

**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.
- Optional: *BulkEditOptionsDTO bulkEditOptions* - DTO containing information required to perform the Apply Bulk Edit Operation.

**Table 1  BulkEditOptionsDTO**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>sourceRules</td>
<td>A List of Source Assignment Rule names being selected for this Bulk Edit Operation</td>
<td>Apply All Building Activities</td>
</tr>
</tbody>
</table>
| destinationRules | A List of Destination Assignment Rule names being selected for this Bulk Edit Operation,  
                  | **Note:** This value should be passed only with BulkEditOperations.ADD_ASSIGNMENT_RULES and BulkEditOperations.REMOVE_ASSIGNMENT_RULES. | 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 BulkEditOperations.ADD_DRIVER operation; however, a list of Driver names can be provided when using the BulkEditOperations.REMOVE_DRIVERS operation | DRV Build Product |
| operation      | Specify the Bulk Edit operation:  
                  | - ADD_DRIVERS  
                  | - REMOVE_DRIVERS  
                  | - ADD_ASSIGNMENT_RULES  
                  | - REMOVE_ASSIGNMENT_RULES | ADD_DRIVERS |
### povGrp

Specify dimension members names for the POV for which this Bulk Edit operation is to be applied:
- povDimensionMember1
- povDimensionMember2
- povDimensionMember3
- povDimensionMember4
- povState

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>povGrp</td>
<td>Specify dimension members names for the POV for which this Bulk Edit operation is to be applied:</td>
<td>povDimensionMember1 = 2012</td>
</tr>
<tr>
<td></td>
<td></td>
<td>povDimensionMember2 = January</td>
</tr>
<tr>
<td></td>
<td></td>
<td>povDimensionMember3 = Actual Draft</td>
</tr>
</tbody>
</table>

### selectAllRulesOrDriversForDelete

Boolean flag specifying if all the destination assignment rules or drivers should be selected for delete. Valid values are TRUE or FALSE.

**Note:** This value should be passed only with BulkEditOperations.REMOVE_DRIVERS and BulkEditOperations.REMOVE_ASSIGNMENT_RULES.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>selectAllRulesOrDriversForDelete</td>
<td>Boolean flag specifying if all the destination assignment rules or drivers should be selected for delete. Valid values are TRUE or FALSE.</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

### selectEntireStageForDelete

Boolean flag specifying if the entire stage should be selected for delete. Valid values are TRUE or FALSE.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>selectEntireStageForDelete</td>
<td>Boolean flag specifying if the entire stage should be selected for delete. Valid values are TRUE or FALSE.</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

### comment

Specify a comment for this Bulk Edit Operation.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>Specify a comment for this Bulk Edit Operation.</td>
<td>“Bulk Edit Operation to Add Drivers”</td>
</tr>
</tbody>
</table>

### Output Parameters

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

**Note:** Use “getTaskflowStatusByProcessName” on page 16 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.

#### Input Parameters

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

#### Table 2  POVMemberGroupDTO

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>povDimensionMember1</td>
<td>POV Dimension Member Name at Position 1</td>
<td>2012</td>
</tr>
<tr>
<td>povDimensionMember2</td>
<td>POV Dimension Member Name at Position 2</td>
<td>January</td>
</tr>
</tbody>
</table>
Variable | Description | Example
---|---|---
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 16 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.

**Input Parameters**

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

**Table 3** ClearPOVDTO

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
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 |
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 |
### Variable

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>clearRevenueLayer</td>
<td>Boolean flag specifying whether Revenue Layers should be cleared. Valid values are TRUE or FALSE.</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

| pov            | Specify dimension member names of the POV for which this Bulk Edit operation should be applied: | povDimensionMember1 = 2012
|                | povDimensionMember2 = January                                                    | povDimensionMember3 = Actual
|                | povState                                                                       | povState = Draft                                                        |

| povGrp         | Specify dimension member names of the POV for which this Bulk Edit operation should be applied: | povDimensionMember1 = 2012
|                | povDimensionMember2 = January                                                    | povDimensionMember3 = Actual
|                | povState                                                                       | povState = Draft                                                        |

| stages        | Specify dimension member names of the POV for which this Bulk Edit operation should be applied: | stages 1
|               | stages 2                                                                       | stages 3
|               | stages N, stages N = {displayOrder = (int), example: 1; stageName = (string)}  | 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.

### 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 3, “ClearPOVDTO”.

**Output Parameters**

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

**Note:** Use “getTaskflowStatusByProcessName” on page 16 operation to get the status of this CES taskflow name (as it is displayed on the Taskflow Status Summary).

**deleteApplications**

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

**Caution!** Exercise caution when using deleteApplications, 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.

**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.
POV Member Group DTO povDTO - Specify the dimension member names of the POV to which this Bulk Edit operation should be applied. See Table 2, “POV Member Group DTO”.

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 2, “POV Member Group DTO”.

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 checkboxes in the application when you select Calculate, ...".

Input Parameters

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

<table>
<thead>
<tr>
<th>Table 4 CubeDeployOptionsDTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
</tr>
<tr>
<td>------------------------------</td>
</tr>
<tr>
<td>cubeType</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>firstTimeDeployment</td>
</tr>
<tr>
<td>Variable</td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td>updateDatabase</td>
</tr>
<tr>
<td>replaceDatabase</td>
</tr>
<tr>
<td>archiveDataBeforeDeploy</td>
</tr>
<tr>
<td>archiveDataAndReloadAfterDeploy</td>
</tr>
<tr>
<td>deleteDataArchiveAfterReload</td>
</tr>
</tbody>
</table>

**Output Parameters**

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

**Note:** Use “getTaskflowStatusByProcessName” on page 16 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 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:

- GENERAL (For Standard Profitability applications)
- 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:
  - 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 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.

**getPOVs**

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 2, “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.
**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.

**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 5  CalcScriptOptionsDTO**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>povGrp</td>
<td>POV information for which Calculation Script generation and execution should be performed. See Table 2, “POVMemberGroupDTO”.</td>
<td>povDimensionMember1=2012, povDimensionMember2=March, povDimensionMember3=Actual, povState=Draft</td>
</tr>
</tbody>
</table>
| layerName           | Layer name for which Calculation scripts should be generated and executed. Valid Values:  
|                     |   - COST                                                                   | COST                                                                   |
|                     |   - REVENUE                                                                |                                                                         |
| 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                                                   |
| 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 16 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 checkboxes 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>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>clearCalculated</td>
<td>Boolean flag specifying whether previously calculated values should be cleared. Valid values are TRUE or FALSE.</td>
<td>TRUE</td>
</tr>
<tr>
<td>executeCalculations</td>
<td>Boolean flag specifying whether calculations should be executed as part of this operation. Valid values are TRUE or FALSE.</td>
<td>TRUE</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> When the executeCalculations flag is set to TRUE, you must provide values for createContributionDetail, createDriverTables, and runSingleCalcRuleSequence.</td>
<td></td>
</tr>
<tr>
<td>createContributionDetail</td>
<td>Boolean flag specifying contribution detail should be created. Valid values are TRUE or FALSE.</td>
<td>TRUE</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>createDetailCalculatedDriverTables</td>
<td>Boolean flag specifying whether calculated driver tables should be created.</td>
<td>TRUE</td>
</tr>
<tr>
<td>dataPOVMemberGroups</td>
<td>List of POV Dimension Member Group details that should be considered for Data POV when calculating. See Table 2, “POVMemberGroupDTO”.</td>
<td>povDimensionMember1=2012 povDimensionMember2=January povDimensionMember3=Actual</td>
</tr>
<tr>
<td>modelPOVMemberGroup</td>
<td>Model POV dimension member group details when performing calculations.</td>
<td>povDimensionMember1=2012 povDimensionMember2=January povDimensionMember3=Actual</td>
</tr>
<tr>
<td>runSingleCalcRuleSequence</td>
<td>Boolean flag specifying whether a single calculation rule sequence should be considered when calculating. Valid values are TRUE or FALSE.</td>
<td>FALSE</td>
</tr>
<tr>
<td>postScript</td>
<td>Name of the post-calculation script</td>
<td>POST</td>
</tr>
<tr>
<td>preScript</td>
<td>Name of the pre-calculation script</td>
<td>PRE</td>
</tr>
<tr>
<td>calcRuleSequence</td>
<td>Calculation Rule Sequence Value.</td>
<td>Valid values are 1 - 9999.</td>
</tr>
<tr>
<td>transferToSrcStgDb</td>
<td>Boolean flag specifying whether data transfer should happen to Source Stage Database.</td>
<td>Valid values are TRUE or FALSE.</td>
</tr>
<tr>
<td>transferToDstStgDb</td>
<td>Boolean flag specifying whether data transfer should happen to Destination Stage Database.</td>
<td>Valid values are TRUE or FALSE.</td>
</tr>
<tr>
<td>transferToContribDb</td>
<td>Boolean flag specifying whether data transfer should happen to Contribution Database.</td>
<td>Valid values are TRUE or FALSE.</td>
</tr>
<tr>
<td>abortOnError</td>
<td>Valid values are TRUE or FALSE.</td>
<td>TRUE</td>
</tr>
<tr>
<td>comment</td>
<td>(string)</td>
<td>optional field</td>
</tr>
</tbody>
</table>

**Output Parameters**

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

**Note:** Use “getTaskflowStatusByProcessName” on page 16 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>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>layerName</td>
<td>Layer name for which the genealogy execution paths should be performed. The following are valid values:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>o COST</td>
<td>Layer.COST</td>
</tr>
<tr>
<td></td>
<td>o REVENUE</td>
<td></td>
</tr>
<tr>
<td>paths</td>
<td>List of genealogy execution paths</td>
<td>1-3-5</td>
</tr>
<tr>
<td>povGrp</td>
<td>Specify the POV dimension member group information pertaining to this genealogy paths execution. See Table 2, &quot;POVMemberGroupDTO&quot;.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>povDimensionMember1=2012</td>
<td></td>
</tr>
<tr>
<td></td>
<td>povDimensionMember2=January</td>
<td></td>
</tr>
<tr>
<td></td>
<td>povDimensionMember3=Actual</td>
<td></td>
</tr>
<tr>
<td></td>
<td>povState=Draft</td>
<td></td>
</tr>
</tbody>
</table>

Output Parameters

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

**Note:** Use “getTaskflowStatusByProcessName” on page 16 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.

**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 7
Output Parameters

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

Note: Use “getTaskflowStatusByProcessName” on page 16 operation to get the status of this CES taskflow name (as it is displayed on the Taskflow Status Summary).

runImportFromStaging

Use this operation to invoke the selected import from Staging tables 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 run.

Output Parameters

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

Note: Use “getTaskflowStatusByProcessName” on page 16 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 8, “Profitability and Cost Management Custom Script Requirements”.

21
Table 8  Profitability and Cost Management Custom Script Requirements

<table>
<thead>
<tr>
<th>Script Items</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Namespace and Location** | Use the same Namespace and Location for every command or operation, including Services:  
  - **Namespace** identifies the Profitability web services: [http://profitability.webservices.epm.oracle](http://profitability.webservices.epm.oracle)  
  - **Location** identifies the location of the ProfitabilityService.wsdl 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:  
  - 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, **String** 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 .wsdl 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 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 23 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 Web Service API Reference - Profitability Services (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 9  Required Files for Sample Client

<table>
<thead>
<tr>
<th>Source Folder</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>%EPM_ORACLE_HOME%/../user_projects/domains/EPMSystem/config/fmwconfig</td>
<td>jps-config.xml</td>
</tr>
<tr>
<td>%EPM_ORACLE_HOME%/../user_projects/domains/EPMSystem/config/fmwconfig</td>
<td>&lt;&lt;Associated keystore file.XXXX.jks&gt;&gt;</td>
</tr>
<tr>
<td>If you are using file-based security, %EPM_ORACLE_HOME%/../user_projects/domains/EPMSystem/config/fmwconfig</td>
<td>cwallet.sso</td>
</tr>
</tbody>
</table>

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 parameters
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

## These next three values are needed only when the security policy is USERNAME_TOKEN
#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
```
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.

From a command or shell window, set the following environment variables:

<table>
<thead>
<tr>
<th>Table 10 Sample Client Environment Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment Variable</td>
</tr>
<tr>
<td>JAVA_HOME</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>MIDDLEWARE_HOME</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

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.

Use the format and operations specified in the sample client file to build your custom script. See “Using the Sample Client File” on page 25.

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 23.

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
```
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.

1. **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.