Oracle® Business Intelligence Web Services Guide

Version 10.1.3.2

December 2006

ORACLE

Part Number: B31769-01

Copyright © 2006, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States 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, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of Oracle BI Web Services

Chapter 3: Description of Structures in Oracle BI Web Services

Chapter 4: Description of Services in Oracle BI Web Services

1 What’s New in This Release

Oracle Business Intelligence Enterprise Edition consists of components that were formerly available from Siebel Systems as Siebel Business Analytics Platform, with a number of significant enhancements.

The *Oracle Business Intelligence Web Services Guide* is part of the documentation set for Oracle Business Intelligence Enterprise Edition. This guide contains reference information for the services and methods in Oracle BI Web Services. This guide contains new material and material that was previously published under the title *Siebel Analytics Web Services Guide*.

Oracle recommends reading the Oracle Business Intelligence Enterprise Edition Release Notes before installing, using, or upgrading the Oracle BI Infrastructure. The Oracle Business Intelligence Enterprise Edition Release Notes are available:

- On the Oracle Business Intelligence Enterprise Edition CD-ROM.
- On the Oracle Technology Network at http://www.oracle.com/technology/documentation/bi_ee.html (to register for a free account on the Oracle Technology Network, go to http://www.oracle.com/technology/about/index.html).

What’s New in Oracle Business Intelligence Web Services Guide, Version 10.1.3.2

Table 1 lists changes described in this version of the documentation to support Version 10.1.3.2 of the software.

Table 1. New Product Features in Oracle Business Intelligence Web Services Guide, Version 10.1.3.2

Description	Topic
New structure.	AccountsFilter Structure on page 16
New structure.	AuthResult Structure on page 17
Updated method.	cancelQuery() Method on page 72
Updated with new Oracle product names and terminology.	Entire Book
New structure.	ErrorInfo Structure on page 18
New structure.	ErrorInfo Structure on page 18
New method.	executeIBotNow() Method on page 39

Table 1. New Product Features in Oracle Business Intelligence Web Services Guide, Version 10.1.3.2

Description	Topic
Removed method getResults(). Replaced with executeSQLQuery() Method on page 72 and executeXMLQuery() Method on page 73.	executeSQLQuery() Method on page 72 and executeXMLQuery() Method on page 73
New structure.	ExportImportFlags Structure on page 19
New method.	getCatalogAccountsDatabase() Method on page 53
New method.	getPermissions() Method on page 54
New method.	GetSessionEnvironment() Method on page 51
New service.	iBotService Service on page 39
Updated method.	_import() Method on page 44
New section.	Oracle BI Web Services Licensing and Permissions on page 12
New structure.	QueryResults Structure on page 22
New method.	readObjects() Method on page 62
Updated method.	removeFolder() Method on page 63
New method.	renameAccount() Method on page 55
Removed method purgeLog(). Log expiration is now controlled by a settings in the instanceconfig.xml file.	ReplicationService Service on page 44
New method.	setItemAttributes() Method on page 63
New structure.	UpdateACLMode Enumeration on page 31
Updated definitions of the updateFlag field.	UpdateACLParams Structure on page 30
New method.	updateCatalogItemACL Method on page 65
New structure.	UpdateCatalogItemACLParams Structure on page 31
New method.	writeObjects() Method on page 66
New structure.	XMLQueryExecutionOptions Structure on page 32
New service.	XMLViewService Service on page 71

2 Overview of Oracle BI Web Services

This chapter provides an overview of Oracle BI Web Services, and contains the following sections:

- What is the Simple Object Access Protocol? on page 9
- What is Oracle BI Web Services? on page 9
- Using Item Signatures in Oracle BI Web Services on page 10
- Accessing the Oracle BI Web Services Interface on page 11
- Oracle BI Web Services Licensing and Permissions on page 12

What is the Simple Object Access Protocol?

The Simple Object Access Protocol (SOAP) is a World Wide Web Consortium (W3C) recommendation for an XML protocol for exchanging information on the Web.

What is Oracle BI Web Services?

Oracle BI Web Services is an application programming interface (API) that implements SOAP. Oracle BI Web Services allows you to perform three types of function:

- Extract results from Oracle BI Presentation Services and deliver them to external applications.
- Perform Oracle BI Presentation Catalog management functions.
- Execute Oracle Business Intelligence alerts (known as iBots).

Oracle BI Web Services allows external applications such as J2EE and .NET to use Oracle Business Intelligence as an analytical calculation and data integration engine. It provides a set of Presentation Services that allow external applications to communicate with Oracle BI Presentation Services. You can use Oracle BI Web Services to extract results from Oracle BI Presentation Services and deliver them to external applications and Web application environments. You can reference a saved report or send the criteria for the report to Oracle BI Web Services.

The formal definition of services and methods in Oracle BI Web Services can be retrieved in WSDL (Web Services Definition Language) format. Proxy classes for the services can be generated automatically.

The XML Schema Definition (XSD) file for the services is the file SawServices.xsd, which is located in the \Web\App\Res\Wsdl\Schemas directory in the Oracle Business Intelligence Platform installation directory. The XSD file is used internally and cannot be used separately. You can access the WSDL document at the following Oracle BI Web Services URL:

```
http://<somehost>/analytics/saw.dll?WSDL
```

Oracle BI Web Services is supported on Oracle JDeveloper, Apache Axis, and the Microsoft .NET framework.

For more information about the SOAP recommendation, consult a reference such as the Microsoft Developer Network or the W3C Web site.

Using Item Signatures in Oracle BI Web Services

Each object has its own signature. Signatures are used in conjunction with writing objects. You need to use the appropriate signature when writing objects. The signatures used by the various methods are provided in the method descriptions given in this guide.

The following example code writes a generic object to set the signatures.

```
If (signature == "queryitem1")
{
   ws.writeReport(o,name,true,true,session);
}
else if (signature == "dashboarditem1")
{
   ws.writeDashboard(o,name,true,true,session);
}
else if (signature == "dashboardpageitem1")
{
   ws.writeDashboardPage(o,name,true,true,session);
}
else if (signature == "globalfilteritem1")
{
   ws.writeDashboardPrompt(o,name,true,true,session);
}
else if (signature == "filteritem1")
{
   ws.writeSavedFilter(o,name,true,true,session);
}
else if (signature == "COXmlDocument1")
{
   ws.writeObject(o,name,true,true,session);
}
else
{
   ws.writeObject(o,name,true,true,session);
}
```

Accessing the Oracle BI Web Services Interface

You can access the Oracle BI Web Services interface on any platform on which a SOAP client library and tools are available. The steps to access the SOAP services are different for each programming environment.

Example of Accessing Oracle BI Web Services from Microsoft Visual Studio

The following procedure shows the steps required to access Oracle BI Web Services from Microsoft Visual Studio.

To access Oracle BI Web Services from Microsoft Visual Studio

1. Open your project in Microsoft Visual Studio.
2. In the Solution Explorer, expand the solution node, right-click References, and choose Add Web Reference.

   The Add Web Reference dialog box is displayed.
3. In the URL field, type the URL to access the Oracle BI Web Services WSDL document.

   The following URL is an example URL to access the Oracle BI Presentation Services WSDL document:

   http://<*somehost*>/analytics/saw.dll?WSDL
4. Click Go.

   The found services and methods are displayed in the Add Web Reference dialog.
5. Click the Add Reference button.

   The Add Web Reference dialog box closes, and the node that represents the added Web reference is displayed in the Solution Explorer pane.
6. To see the added classes and methods, right-click the node and choose the following option:

   View in Object Browser

   The classes and methods are displayed in the Object Browser window.
7. Begin using the classes in your program.

   For a code example, see “Code Example” on page 77.

Oracle BI Web Services Licensing and Permissions

Oracle BI Web Services is available to licensed users of Oracle Business Intelligence. The installer generates the appropriate licensing entries in the analyticsweblicense.xml file based on the installation key. If you get a “Not Licensed” error when making an Oracle BI Web Services method, then check that you used the correct key at installation.

The licensing entries in the analyticsweblicense.xml file are:

- kmsgLicenseSOAPAccess. This entry enables the SOAP interfaces.
- kmsgLicenseOfficeIntegration. This entry enables integration with Microsoft Excel.

The Access Soap permission is granted to all users by default. If you explicitly deny this permission to a user, then Oracle BI throws an "Access Denied" exception for Oracle BI Web Services methods that require authentication (examples: logon and logonex).

3 Description of Structures in Oracle BI Web Services

This chapter describes the structures used by Oracle BI Web Services.

NOTE: This document uses JavaScript-like syntax to describes structures. The exact syntax and implementation depends on the SOAP code generation tool and the target language used by your application development environment.

This chapter contains the following sections:

- UpdateACLParams Structure on page 30
- UpdateCatalogItemACLParams Structure on page 31
- Variable Structure on page 32
- XMLQueryExecutionOptions Structure on page 32

Structures and Services

Table 2 lists structures, grouped by the services that use them.

Table 2. Services and Their Applicable Structures

Services	Structures
All services	“ErrorInfo Structure” on page 18
	“ReportParams Structure” on page 24
	“ReportRef Structure” on page 25
	“Variable Structure” on page 32
HtmlViewService	“ReportHTMLOptions Structure” on page 23
	“StartPageParams Structure” on page 30
MetadataService	“SAColumn Structure” on page 25
	“SASubjectArea Structure” on page 27
	“SATable Structure” on page 28
ReplicationService	“CatalogItemsFilter Structure” on page 17
	“ExportImportFlags Structure” on page 19
	“ImportError Structure” on page 20
SAWSessionService	“AuthResult Structure” on page 17
	“SAWLocale Structure” on page 28
	“SAWSessionParameters Structure” on page 29
SecurityService	“AccessControlToken Structure” on page 15
	“ACL Structure” on page 16
	“AccountsFilter Structure” on page 16
	“Privilege Structure” on page 22
	“UpdateACLParams Structure” on page 30
	“UpdateCatalogItemACLParams Structure” on page 31

Table 2. Services and Their Applicable Structures

Services	Structures
WebCatalogService	"CatalogObject Structure" on page 18
	"ErrorInfo Structure" on page 18
	"GetSubItemsParams Structure" on page 19
	"ItemInfo Structure" on page 21
	"NameValuePair Structure" on page 22
	"NameValuePair Structure" on page 22
	"UpdateCatalogItemACLParams Structure" on page 31
XMLViewService	"QueryResults Structure" on page 22
	"ReportParams Structure" on page 24
	"ReportRef Structure" on page 25
	"XMLQueryExecutionOptions Structure" on page 32

AccessControlToken Structure

Use this structure to describe permissions granted to a specific account in the access control list. This structure is used in the "SecurityService Service". Table 3 lists the fields in this structure.

Table 3. AccessControlToken Structure Fields

Fields	Description
Account account	Specifies a reference to the Account structure.
int permissionMask	Specifies a combination of the following flags: 1 = Permission to read item content 2 = Permission to traverse directory 4 = Permission to change item content 8 = Permission to delete an item 16 = Permission to assign permissions to other accounts 32 = Permission to take ownership of the item

Account Structure

Use this structure to hold user names or group names. It has a flag to indicate whether the name is a user or a group. This structure is used in the "SecurityService Service". Table 4 lists the fields in this structure.

Table 4. Account Structure Fields

Fields	Description
String accountName	Specifies an account name or group name.
int accountType	Specifies whether the account is a user or a group (0 = user, 1 = group).

AccountsFilter Structure

Use this structure to specify how you filter accounts in the cache. This structure is used in the "SecurityService Service" (in the getCatalogAccountsDatabase method). Table 5 lists the fields in this structure.

Table 5. AccountsFilter Structure Fields

Fields	Description
boolean includeUsers	If set to TRUE, then filter the accounts in the cache by user account.
boolean includeGroups	If set to TRUE, then filter the accounts in the cache by user group.

NOTE: You can use includeUsers and includeGroups in the same command.

ACL Structure

Use this structure to hold the access control list (ACL). This structure is used in the "SecurityService Service". Table 6 lists the fields in this structure.

Table 6. ACL Structure Fields

Fields	Description
AccessControlToken[] accessControlTokens	Specifies the full list of permissions.
Account owner	Specifies the owner of the resource.

AuthResult Structure

Use this structure to specify authorization details during an authentication. This structure is used in the "SAWSessionService Service" (in the "logonex() Method" and "impersonateex() Method"). Table 7 lists the fields in this structure.

Table 7. AuthResult Structure Fields

Fields	Description
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.
boolean authCompleted	If set to TRUE, then the authorization is complete. If set to FALSE, then the authorization process is in progress and the logonex or impersonatex process should be called again.

CatalogItemsFilter Structure

Use this structure to filter catalog items and changes based on the path and timestamp. This structure is used in the "ReplicationService Service". Table 8 lists the fields in this structure.

Table 8. CatalogItemsFilter Structure Fields

Fields	Description
String[] items	Specifies the list of folders and their descendants to include in the filter. If this value is null, then all nodes in the catalog are included.
Calendar from	Specifies the time period on which to filter. Only items and changes with timestamps within that period satisfy the filter (from <= timestamp <= to). Either or both of those fields could be null, in which case corresponding bound is considered not set.
Calendar to	

CatalogObject Structure

Use this structure to retrieve or specify all information for a particular Presentation Catalog object in a single method. This structure is used in the "WebCatalogService Service". Table 9 lists the fields in this structure.

Table 9. CatalogObject Structure Fields

Fields	Description
String catalogObject	Specifies an XML representation of the object.
ItemInfo itemInfo	Specifies Presentation Catalog information about the object, supplied in the ItemInfo common structure. For information about the ItemInfo structure, read "ItemInfo Structure" on page 21.
ErrorInfo errorInfo	Specifies the level of error information to be supplied as specified by the ErrorDetails argument in the readObjects method (for more information, see "readObjects() Method" on page 62).

ErrorInfo Structure

Use this structure to retrieve error information during Presentation Catalog Service method invocations. This structure is used in the "WebCatalogService Service". Table 10 lists the fields in this structure.

Table 10. ErrorInfo Structure Fields

Fields	Description
String code	Specifies the error code to display.
String context	Specifies the service and method in which the error occurred.
String details	Specifies detailed information about the error.
String message	Specifies a human-readable description of the error.

ExportImportFlags Structure

Use this structure to specify the changes to export during export methods. This structure is used in the "ReplicationService Service". Table 11 lists the fields in this structure.

NOTE: Only one of the fields in ExportImportFlags should be populated.

Table 11. ExportImportFlags Structure Fields

Fields	Description
String processAll	Specifies that you want to export all items in the directory as if they were inserted, without referring to replication logs.
String processAllChanges	Specifies that you want to export flags that were changed in a given directory on the LOCAL computer and on REMOTE computers, after first referring to the replication logs to find out what changes were made (combines processLocalChanges and processRemoteChanges).
String processLocalChanges	Specifies that you want to export flags that were changed in a given directory on the LOCAL computer, after first referring to the replication logs to find out what changes were made.
String processRemoteChanges	Specifies that you want to export flags that were changed in a given directory on REMOTE computers, after first referring to the replication logs to find out what changes were made.

GetSubItemsParams Structure

Use this structure to contain optional parameters used in a getSubItems method. This structure is used in the "WebCatalogService Service". Table 12 lists the fields in this structure.

Table 12. GetSubItemsParams Structure Fields

Fields	Descriptions
NameValuePair	Used for internal purposes only. This field should be null.
boolean includeACL	If set to TRUE, then ACL information is included in the resulting ItemInfo structures.
int withPermission	Specifies that you want to filter the resulting items collection by access level. The only items included in the result are those for which the following expression is true: `(itemPermission & withPermissionMask) = (withPermission & withPermissionMask)` Where itemPermission is a combination of permission flags for current catalog item.
int withPermissionMask	

Table 12. GetSubItemsParams Structure Fields

Fields	Descriptions
int withAttributes	Specifies that you want to filter the resulting items collection by attribute flags. The only items included in the result are those for which the following expression is true: `(itemAttributes & withAttributesMask) = (withAttributes & withAttributesMask)` Where itemAttributes is a combination of attribute flags for current catalog item.
int withAttributesMask	

ImportError Structure

Use this structure to describe the cause of a failure during an import. This structure is used in the "ReplicationService Service". Table 13 lists the fields in this structure.

Table 13. ImportError Structure Fields

Fields	Description
String item	Specifies the path to the changed item. For example, /users/jchan/reports/.
String operation	For internal use only.
String file	
int line	
String catalogError	Specifies an error string, describing the reason for the failure.

ItemInfo Structure

Use this structure to contain Presentation Catalog information about an object. This structure is used in the "WebCatalogService Service". Table 14 lists the fields in this structure.

Table 14. ItemInfo Structure Fields

Fields	Description
String path	Specifies the path to the object in the Presentation Catalog. For example, /users/jchan/reports/.
ItemInfoType type	Specifies a character string that indicates the type. Valid values are: ■ Folder ■ Link ■ Missing ■ NoAccess ■ Object
String caption	Specifies the localized name of the object in the Presentation Catalog. For example, in French, 'My Folders' is displayed as 'Mes Dossiers'.
int attributes	Specifies a combination of the following flags: 1 = read only 2 = archive 4 = hidden 8 = system
Calendar lastModified	Specifies the date and time that the object was last modified, in Calendar format.
Calendar created	Specifies the date and time that the object was created (saved) in the Presentation Catalog, in Calendar format.
Calendar accessed	Specifies the data and time that the object was last accessed by a user, in Calendar format.
String signature	Specifies the signature of the Web Catalog object. For more information about signatures, read "Using Item Signatures in Oracle BI Web Services" on page 10.
NameValuePair[] itemProperties	Specifies an array of object properties.
ACL acl	Specifies the Access Control List for this catalog item.

NameValuePair Structure

Use this structure to denote named properties, such as COLOR=RED. This structure is used in the "WebCatalogService Service". Table 15 lists the fields in this structure.

Table 15. NameValuePair Structure Fields

Fields	Description
String name	Specifies a character string that contains the name of the property, such as COLOR.
String value	Specifies a character string that contains the value, such as RED.

Privilege Structure

Use this structure to represent global privileges. In Oracle Business Intelligence, you configure these privileges using the Manage Privileges screen. This structure is used in the "SecurityService Service". Table 16 lists the fields in this structure.

Table 16. Privilege Structure Fields

Fields	Description
String name	Specifies the name of a privilege.
String description	Specifies the description of a privilege.

QueryResults Structure

Use this structure to specify query details during query execution.This structure is used in the "XMLViewService Service" (in the executeXMLQuery method). Table 17 lists the fields in this structure.

Table 17. QueryResults Structure Fields

Fields	Description
String rowset	Specifies the rowset XML encoded in the string.
String queryID	Specifies the unique ID of the query, which can be used in fetchNext calls.
boolean finished	If set to TRUE, then there are no more rows to return. If set to FALSE, then another fetchNext call is needed to return more rows.

ReportHTMLOptions Structure

Use this structure to define options for displaying results on an HTML page. This structure is used in the "HtmlViewService Service". Table 18 lists the field in this structure.

Table 18. ReportHTMLOptions Structure Field

Field	Description
boolean enableDelayLoading	Internal use only. This field is always set to 1, which means that Oracle BI Web Services is never required to provide results immediately, and displays a message indicating that it is waiting for results.
String ReportHTMLLinksMode	Specifies whether to display drills or links in the current browser window or a new browser window. For more information about valid values, see ReportHTMLLinksMode Enumeration on page 23).

ReportHTMLLinksMode Enumeration

This enumeration specifies a list of valid values for the ReportHTMLLinksMode field in the ReportHTMLOptions Structure. Table 19 lists the values in this enumeration.

Table 19. ReportHTMLLinksMode Enumeration Values

Values	Description
String InPlace	Specifies that drills or links should replace only the content of the current report without changing the rest of the page.
String NewPage	Specifies that drills or links should be displayed in a new browser window.
String SamePage	Specifies that drills or links should replace the current browser window.

ReportParams Structure

Use this structure to replace existing filters and variables in a report. This structure is common to all services in Oracle BI Web Services. Table 20 lists the fields in this structure.

Table 20. ReportParams Structure Fields

Fields	Description
String[] filterExpressions	Specifies an array of Oracle BI Web Services filter expressions in the form Object[] filter_expression, filter_expression ...
Variable[] variables	Specifies an array of variable values to be set before method execution. This structure is used in the “executeXMLQuery() Method”, the “cancelQuery() Method”, and the “generateReportSQL() Method”.
NameValuePair[] nameValues	Should be set to NULL. This field is for internal use only.
TemplateInfo[] templateInfos	Should be set to NULL. This field is for internal use only.

Table 21 shows how filter expressions are applied to a report.

Table 21. How Filter Expressions Are Applied to a Report in Oracle BI Web Services

Step	Internal Processing
1	Obtains XML representations of the report and each filter expression.
2	For each expression element, locates the child node of the type sqlExpression (the type is determined by the value of the xsi:type attribute), and references its inner text.
3	In the report XML, locates all nodes that also have a child node of type sqlExpression where the inner text matches that located in the preceding step.
4	Replaces all nodes found in Step 3 with the expression from Step 2.

Table 22 shows how variables are applied to a report.

Table 22. How Variables Are Applied to a Report in Oracle BI Web Services

Step	Internal Processing
1	Obtains XML representations of the report.
2	For each variable, locates all nodes in the report XML that have a type of variable, attribute scope equal to report, and inner text that matches the variable name.
3	Replaces each node located in Step 2 with the new variable value.

ReportRef Structure

Use this structure to reference a report, in one of the following ways:

- The location of the report in the Web Catalog.
- The ReportDef object that defines the report. This field should always be null.
- The XML that defines the report.

NOTE: Only one of the fields in ReportRef should be populated.

The ReportRef structure is common to all services in Oracle BI Web Services. Table 23 lists the fields in this structure.

Table 23. ReportRef Structure Fields

Fields	Description
String reportPath	Specifies a string value that provides the path to the report in the Presentation Catalog. For example, /users/jchan/reports/.
String reportXML	Specifies a string value that contains the XML that defines the report.

SAColumn Structure

Use this structure to represent the logical column in the Subject Area. This structure is used in the "MetadataService Service". Table 24 lists the fields in this structure.

Table 24. SAColumn Structure Fields

Fields	Description
String name	Specifies a column name used in SQL statements.
String displayName	Specifies a localized name, used in Oracle BI Answers.
String description	Specifies a string to contain the description of the column name.
boolean nullable	If set to TRUE, then the column can be null.
String dataType	Specifies the type of data that a column contains. For more information, read SADataType Values on page 26.
boolean aggregateable	If set to TRUE, then the column can be aggregated.
String aggrRule	If the column contains aggregated data, this value specifies the type of aggregation used. For more information, read AggregationRule Values on page 26.

SADataType Values

The SADataType indicates the type of data that a column contains. The following list shows the data types available:

- BigInt
- Binary
- Bit
- Char
- Coordinate
- Date
- Decimal
- Double
- Float
- Integer
- Invalid
- LongVarBinary
- LongVarChar
- Numeric
- Real
- SmallInt
- Time
- TimeStamp
- TinyInt
- Unknown
- VarBinary
- VarChar

AggregationRule Values

The SADataType specifies the default aggregation rule for the column. For details on aggregation functions, read *Oracle Business Intelligence Presentation Services Administration Guide*. The following list shows the aggregation functions available:

- Avg
- BottomN
- Complex
- Count

- CountDistinct
- CountStar
- DimensionAggr
- First
- Last
- Max
- Min
- None
- Percentile
- Rank
- ServerDefault
- SubTotal
- Sum
- TopN

SASubjectArea Structure

Use this structure to represent Subject Area attributes. This structure is used in the "MetadataService Service". Table 25 lists the fields in this structure.

Table 25. SASubjectArea Structure Fields

Fields	Description
String name	Specifies the table name that is used in SQL statements.
String displayName	Specifies the localized name, used in Oracle BI Answers.
String description	Specifies the description of the subject area.
SATable[] tables	Specifies a collection of tables for this subject area. For information about the SATable structure, read "SATable Structure" on page 28.

SATable Structure

Use this structure to represent the logical table in the Subject Area. This structure is used in the "MetadataService Service". Table 26 lists the fields in this structure.

Table 26. SATable Structure Fields

Fields	Description
String name	Specifies the table name that is used in SQL statements.
String displayName	Specifies the localized name, used in Oracle BI Answers.
String description	Specifies the description of the table name.
SAColumn[] columns	Specifies an array of the table's columns. For information about the SAColumn structure, read "SAColumn Structure" on page 25.

SAWLocale Structure

Use this structure to define the locale for the current session. This structure is used in the "SAWSessionService Service". Table 27 lists the fields in this structure.

Table 27. SAWLocale Structure Fields

Fields	Description
String language	Specifies the language code. Values for language should conform to the ones used in Java, in the java.util.Locale class (ISO-639, ISO-3166).
String country	Specifies the country code. Values for country should conform to the ones used in Java, in the java.util.Locale class (ISO-639, ISO-3166).

SAWSessionParameters Structure

Use this structure to define optional parameters for the current session. This structure is used in the "SAWSessionService Service". Table 28 lists the fields in this structure.

Table 28. SAWSessionParameters Structure Fields

Fields	Description
SAWLocale locale	Specifies the locale to be used, supplied in the SAWLocale structure. For information about the SAWLocale structure, read "SAWLocale Structure" on page 28.
String userAgent	Specifies whether the HTMLView service will be used with current session. It specifies the userAgent string of the browser, where Oracle BI Presentation Services HTML content is displayed. Oracle BI Presentation Services uses this information to produce browser-specific HTML.
String features	For internal use only. Should be null.
boolean asyncLogon	If set to TRUE, then asynchronous login is enabled. If set to FALSE (default), then asynchronous login is not enabled.
String sessionID	Specifies the unique ID of the session. This field is used in the "logonex() Method" and impersonateex() Method.

SessionEnvironment Structure

Use this structure to return environment information for the current session. This structure is used in the "SAWSessionService Service". Table 29 lists the fields in this structure.

Table 29. SessionEnvironment Structure Fields

Fields	Description
String userName	Specifies the name of the current user.
ItemInfo homeDirectory	Specifies the full path to the user's home directory in the Presentation Catalog. For example, /users/<*user login ID*>.
ItemInfo[] SharedDirectories	Specifies the full paths to shared directories to which the current user has at least read access. **NOTE:** By default, only administrators are allowed to list direct descendents of the "/shared" directory. Retrieving the SessionEnvironment object is the only way to enable users to navigate its shared area.

StartPageParams Structure

Use this structure to define options in startPage method invocations. This structure is used in the "HtmlViewService Service". Table 30 lists the fields in this structure.

Table 30. StartPageParams Structure Fields

Fields	Description
String idsPrefix	Specifies a prefix to be used with IDs and names of all HTML elements to avoid name conflicts on an HTML page.
boolean dontUseHttpCookies	If set to TRUE, then Oracle BI Presentation Services cannot rely on cookies for passing the sessionID. Instead, the sessionID is included as a parameter in callback URLs.

UpdateACLParams Structure

Use this structure to set options in updateACL method invocations. This structure is used in the "SecurityService Service". Table 31 lists the fields in this structure.

See also UpdateACLMode Enumeration on page 31.

Table 31. UpdateACLParams Structure Fields

Fields	Description
boolean allowUnknownAccounts	If set to TRUE (1), and the new access control list (ACL) in the updateACL includes accounts that are unknown to Oracle BI Web Services, then Oracle BI Web Services creates new account records for them. However, to be used, these accounts should exist in Oracle BI Server as well.
UpdateACLMode int updateFlag	Specifies how to update the ACL, as follows: 0 = Replace the existing ACL with the new one. 1 = Merge the new ACL with the existing one. 2 = Revoke privileges. The new ACL contains a list of accounts and privileges to be revoked.

UpdateACLMode Enumeration

This enumeration specifies a list of valid values for the update flag in the UpdateACLParams Structure. Table 32 lists the values in this enumeration.

Table 32. UpdateACLMode Enumeration Values

Values	Description
String ReplaceACL	Specifies the ACL value to update.
String ReplaceForSpecifiedAccounts	Specifies a list of accounts to update in the ACL.
String DeleteAccountsFromACL	Specifies a list of accounts to remove from the ACL.

UpdateCatalogItemACLParams Structure

Use this structure to provide additional parameters in the updateCatalogItemACL Method on page 65. This structure is used in the “WebCatalogService Service”. Table 33 lists the fields in this structure.

Table 33. UpdateCatalogItemACLParams Structure Fields

Fields	Description
boolean allowUnknownAccounts	If set to TRUE (1), and the new access control list (ACL) in the updateACL includes accounts that are unknown to Oracle BI Web Services, then Oracle BI Web Services creates new account records for them. However, to be used, these accounts should exist in Oracle BI Server as well.
UpdateACLMode updateFlag	Specifies how to update the ACL, as follows: 0 = Replace the existing ACL with the new one. 1 = Merge the new ACL with the existing one. 2 = Revoke privileges. The new ACL contains a list of accounts and privileges to be revoked.
boolean recursive	If set to TRUE, then the method is applied to the catalog item and all descendents, which are identified by the path. If set to FALSE, then the method is only applied to the catalog item.

Variable Structure

Use this structure to reference a variable in the report and replace it with another variable. This structure is common to all services in Oracle BI Web Services. Table 34 lists the fields in this structure.

Table 34. Variable Structure Fields

Fields	Description
String name	Specifies a character string that contains the name of the variable to replace.
Object value	Specifies the value of the variable.

XMLQueryExecutionOptions Structure

Use this structure to specify optional parameters during a query. This structure is used in the "XMLViewService Service" (in the executeXMLQuery method). Table 35 lists the fields in this structure.

Table 35. XMLQueryExecutionOptions Structure Fields

Fields	Description
boolean async	If set to TRUE, then asynchronous query execution is enabled. If set to FALSE, then asynchronous query execution is disabled.
int maxRowsPerPage	Specifies the maximum number of rows to be returned by a executeXMLQuery or fetchNext method.
boolean refresh	If set to TRUE, then the server re-submits the query to refresh the data. If set to FALSE, then the Oracle BI Server uses data in the cache.
boolean presentationInfo	If set to TRUE, then store localized presentation information in the metadata section of the record set XML. Presentation information consists of the following: ■ Column heading information (stored in the columnHeading field). ■ Table heading information (stored in the tableHeading field).
String type	Specifies the query ID, which can be used in logs to diagnose errors.

4 Description of Services in Oracle BI Web Services

This chapter describes the services and methods in Oracle BI Web Services.

NOTE: This document uses JavaScript-like syntax to describes structures. The exact syntax and implementation depends on the SOAP code generation tool and the target language used by your application development environment.

This chapter describes the following services:

- HtmlViewService Service on page 33
- iBotService Service on page 39
- MetadataService Service on page 40
- ReplicationService Service on page 44
- ReportEditingService Service on page 46
- SAWSessionService Service on page 47
- SecurityService Service on page 52
- WebCatalogService Service on page 57
- XMLViewService Service on page 71

NOTE: For information about which structures use which services, see Structures and Services on page 14.

HtmlViewService Service

Use this service to embed Oracle BI HTML results in third-party dynamic Web pages, such as Active Server Pages (ASP) or JavaServer Pages (JSP), and portal frameworks. The embed process merges Oracle BI Web Services content with the content of third-party Web pages. Table 36 shows the supported methods.

Table 36. HtmlViewService Methods

Method Name	Description
addReportToPage() Method on page 35	Adds results to an HTML page.
endPage() Method on page 36	Destroys a server page object and all data associated with it.
getCommonBodyHTML() Method on page 36	Retrieves HTML to include in the <BODY> section.
getHeadersHTML() Method on page 36	Retrieves HTML to include in the <HEAD> section.
getHTMLForReport() Method on page 37	Retrieves HTML to display a particular set of results.

Table 36. HtmlViewService Methods

Method Name	Description
setBridge() Method on page 38	Specifies a bridge URL to receive communications. Can be useful when the Oracle BI Web Services server and the user's Presentation Services reside on different machines or when you want to modify the results in your application development environment.
startPage() Method on page 39	Creates a new page object and returns its ID.

The methods in the HTMLViewService service extract fragments of HTML code that can be inserted in third-party Web pages. Table 37 describes the HTML code excerpts and desired page locations.

Table 37. HTML Code Fragments and Page Locations for the HtmlViewService Service

HTML Code Fragment	Desired Page Location
Header	Should be inserted in the <HEAD> section of an HTML page. The code contains links to common JavaScript files and style sheets.
Report Objects	Can be inserted anywhere in the <BODY> section.
Common Body	Should be inserted in the <BODY> tag after all report links. The code contains hidden HTML elements that are used to implement drilldown links.

For each returned report object, the HTML code fragment contains a callback link that is followed automatically when the Web page is loaded by the browser. The code fragment does not contain the full user interface definition of the report. While the report is being constructed by Oracle BI Web Services, the interface displays the Oracle BI Web Services "Searching..." image embedded on the third-party Web page.

For smooth report transitioning, Oracle BI Web Services tracks the Oracle BI reports that have been added to third-party Web pages by maintaining information in an internal logical page object during the construction of the third-party Web page. The HtmlViewService service methods explicitly refer to the internal logical page by its ID.

About HtmlViewService Bridging and Callback URLs

To embed a report with active drilldown links, the HtmlViewService service allows the Web browser to issue callback requests from embedded reports to the Oracle BI Web Services server. Although it is possible to route requests directly to the Oracle BI Web Services server, in many cases it is preferable to route requests through the Presentation Services that originally serviced the third-party page. Also, in situations where Oracle BI Web Services and the third-party Web server do not belong to the same Domain Name Service (DNS) domain, users may get JavaScript errors related to browser security constraints for cross-domain scripting.

To avoid these issues, use the setBridge() method to modify callback URLs to point to the third-party Web server. Be aware that a Web component executed by the third-party Web server to re-route requests to Oracle BI Web Services is not provided. This function would need to be fulfilled by the third-party application. For more information about the setBridge() method, read "setBridge() Method" on page 38.

addReportToPage() Method

Use this method to add results to an HTML page.

Signature

void addReportToPage(String pageID, String reportID, ReportRef report, String reportViewName, ReportParams reportParams, ReportHTMLOptions options, String sessionID);

Arguments	Description
String pageID	Specifies a character string page ID returned by the startPage() method. For information about the startPage () method, read "startPage() Method" on page 39.
String reportID	Specifies a character string that identifies the report containing the results to add to the page. It should be used to reference this report in subsequent method invocations; for example, corresponding user interface elements generated by the Oracle BI Web Services server would reference the same ID.
ReportRef report	Specifies the report definition, supplied in the ReportRef structure. For more information, read "ReportRef Structure" on page 25.
String reportViewName	Specifies the view to display. If this parameter is null, the report's default view is used. The view name should match the one used to identify the view in the report XML definition.
ReportParams reportParams	Optional. Specifies the filters or variables to apply to the report before execution, supplied in the ReportParams common structure. For more information, read "Description of Services in Oracle BI Web Services" on page 33.
ReportHTMLOptions options	Optional. Specifies the display options to apply to the report after execution, supplied in the ReportHTMLOptions structure. For more information, read "QueryResults Structure" on page 22.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

endPage() Method

Use this method to destroy the Oracle BI Web Services server page object and all data associated with it.

Signature

void endpage(String pageID, String sessionID);

Arguments	Description
String pageID	Specifies the ID of the page object, which is returned by the startPage() method (for more information, read "startPage() Method" on page 39).
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

getCommonBodyHTML() Method

Use this method to retrieve HTML to include in the <BODY> section.

Signature

String getCommonBodyHTML(String pageID, String sessionID);

Arguments	Description
String pageID	Specifies the ID of the page object, which is returned by the startPage() method (for more information, read "startPage() Method" on page 39).
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns a string containing the HTML to include in the <BODY> section.

getHeadersHTML() Method

Use this method to retrieve HTML to include in the <HEAD> section.

Signature

String getHeadersHTML(String pageID, String sessionID);

Arguments	Description
String pageID	Specifies the ID of the page object, which is returned by the startPage() method (for more information, read "startPage() Method" on page 39).
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns a string containing the HTML to include in the <HEAD> section.

getHTMLForReport() Method

Use this method to retrieve an HTML excerpt to display the results for a particular report. Before invoking this method, use the addReportToPage method to add the results to an HTML page.

Signature

String getHTMLForReport(String pageID, String pageReportID, String sessionID);

Arguments	Description
String pageID	Specifies the ID of the page object, which is returned by the startPage() method (for more information, read "startPage() Method" on page 39).
String pageReportID	Specifies the report ID returned by the addReportToPage() method. For information about the addReportToPage method, read "addReportToPage() Method" on page 35.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns a string containing the HTML excerpt that displays the specified report.

setBridge() Method

Use this method to specify a bridge URL to receive communications. Specifying a bridge URL can be useful when the Oracle BI Web Services server and the user's Web server reside on different machines, or when you want to modify the results in your application development environment.

After the setBridge() method is called, all requests from the client browser to the Oracle BI Web Services server are sent to the bridge URL, which then forwards requests to the Oracle BI Web Services server.

Signature

void setBridge(String bridgeURL, String sessionID);

Arguments	Description
String bridgeURL	Specifies the bridge URL. For example, http://myserver/myapplication/sawbridge.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Usage

You must make sure that the client browser provides a handler to the bridge URL in the form of a Java servlet, an Active Server Pages (ASP) page, a Common Gateway Interface (CGI), an Internet Server application programming interface (ISAPI), or an equivalent application.

You must also perform the following tasks:

- Decode the path of the requested Oracle BI Web Services resource in the RedirectURL argument of the request character string. For information about the RedirectURL argument, read "How Callback URLs Are Replaced" on page 38.
- Forward all other request arguments, together with all headers and the request body, to the bridge URL.
- Copy the response from the Oracle BI Web Services server to the response stream.

How Callback URLs Are Replaced

The new callback URL is based on the bridge URL, with the addition of a RedirectURL argument. The value of the RedirectURL argument should be the original value of the URL, encoded using standard URL encoding rules.

Internally, Oracle BI Web Services usually uses relative URLs for callback links. For example, if the original callback link is saw.dll?Go and the bridge URL is http://myserver/myapplication/sawbridge, the new callback URL is http://myserver/myapplication/sawbridge?RedirectURL=saw.dll%3fGo.

startPage() Method

Use this method to create a new page object and returns its ID.

Signature

String startPage(StartPageParams options, String sessionID);

Arguments	Description
StartPageParams options	Specifies the options to use when starting the page, supplied in the StartPageParams structure. For information about the StartPageParams structure, read "StartPageParams Structure" on page 30.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns a string containing the Oracle BI Presentation Services page ID.

iBotService Service

Use this service to execute Oracle BI alerts (known as iBots) programatically. Before you can execute an iBot, you must first create the iBot in Oracle BI Delivers and store it in the Presentation Catalog. Table 38 shows the supported methods.

Table 38. iBotService Methods

Method Names	Description
executeIBotNow() Method on page 39	Executes an iBot that is stored in the Presentation Catalog.

executeIBotNow() Method

Use this method to execute an iBot that is stored in the Presentation Catalog.

Signature

executeIBotNow (String path, String sessionID);

Arguments	Description
String path	Specifies the full path and name of the iBot in the Presentation Catalog. For example, /users/jchan/_ibots/BrandDollars.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

MetadataService Service

Use this service to retrieve descriptions of Oracle BI Web Services schema objects, such as columns, tables, and subject areas. Table 39 shows the supported methods.

Table 39. MetadataService Methods

Method Names	Description
describeColumn() Method on page 40	Retrieves column information for a specified column in a specified subject area and table.
describeSubjectArea() Method on page 41	Retrieves subject area information for a specified subject area.
describeTable() Method on page 42	Retrieves table information for a specified table in a specified subject area.
getSubjectAreas() Method on page 43	Retrieves the list of subject areas available.

describeColumn() Method

Use this method to retrieve column information for a specified column in a specified subject area and table.

Signature

SAColumn describeColumn (String subjectAreaName, String tableName, String columnName, String sessionID);

Arguments	Description
String subjectAreaName	Specifies the subject area to be queried.
String tableName	Specifies the table to be queried.

Arguments	Description
String columnName	Specifies the name of the column to be queried.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns an SAColumn Object. For information on the SAColumn structure, read "SAColumn Structure" on page 25.

describeSubjectArea() Method

Use this method to retrieve subject area information about the specified subject area.

Signature

SASubjectArea describeSubjectArea (String subjectAreaName, SASubjectAreaDetails detailsLevel, String sessionID);

Arguments	Description
String subjectAreaName	Specifies the subject area to be queried.
SASubjectAreaDetails detailsLevel	Specifies the information to be retrieved about the subject area. For information on the SASubjectAreaDetails structure, read "SASubjectAreaDetails Values" on page 41.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

SASubjectAreaDetails Values

Use this method to specify what information should be retrieved about the subject area. Table 40 lists the available values.

Table 40. SASubjectAreaDetails Values

Values	Description
IncludeTables	Include table list with minimum information about each table.
IncludeTablesAndColumns	Include full table and column information.
Minimum	Do not include table and column information.

Returns

Returns an SASubjectArea Object (for more information, read "SASubjectArea Structure" on page 27).

Usage

Depending on the value of the detailsLevel parameter, the returned object contains the information specified in Table 41.

Table 41. detailsLevel Values

Value of detailsLevel	Description
IncludeTables	Specifies that the tables field is not null and contains the collection of tables for this subject area. Each table object has the columns field set to null.
InludeTablesAndColumns	Specifies that the tables field is not null and contains the collection of tables for this subject area. For each table object the columns field contains the corresponding collection of columns.
Minimum	Specifies that the table list is not available. The tables field in the resulting subject area object is null.

describeTable() Method

Use this method to retrieve table information for a specified table in a specified subject area.

Signature

SATable describeTable (String subjectAreaName, String tableName, SATableDetails detailsLevel, String sessionID);

Arguments	Description
String subjectAreaName	Specifies the subject area to be queried.
String tableName	Specifies the table to be queried.
SATableDetails detailsLevel	Specifies the information to retrieve about the table. For information on the SATableDetails structure, read "SATablesDetails Values" on page 43.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

SATablesDetails Values

Used to specify the information to retrieve about the table. Table 42 lists the available values.

Table 42. SATableDetails Values

Values	Description
IncludeColumns	Populate the columns field in the SATable Object.
Minimum	Do not include column information. The columns field in the SATable Object is set to null.

Returns

Returns an SATable Object. For information on the SATable structure, read "SATable Structure" on page 28.

getSubjectAreas() Method

Use this method to retrieve the list of subject areas available.

Signature

SASubjectArea[] getSubjectAreas(String sessionID);

Arguments	Description
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns an array of SASubjectArea objects. For information on the SASubjectArea structure, read "SAWLocale Structure" on page 28.

Usage

SASubjectArea objects returned by this method do not have table information available. The tables field is null. The approach to querying at all levels is to use getSubjectAreas() to retrieve the list of subject areas, then use describeSubjectArea() to retrieve the list of tables. Then use describeTable() to retrieve the list of columns in a specified table, and finally use describeColumn() to retrieve information on a specified column.

ReplicationService Service

Use this service to provide catalog replication methods. Table 43 shows the supported methods.

Table 43. ReplicationService Methods

Method Names	Description
export() Method on page 44	Exports catalog changes to a specified log file.
_import() Method on page 44	Import changes from the log file.
markForReplication() Method on page 45	Change the "replicable" flag on a specified folder and its descendants.

export() Method

Use this method to export catalog changes to a specified log file.

Signature

void export (String filename, CatalogItemsFilter filter, ExportImportFlags flag, String sessionID);

Argument	Description
String filename	Specifies the name of the log file.
CatalogItemsFilter filter	Specifies the subset of changes to be exported. The filter.items field cannot be null.
ExportImportFlags flag	Specifies the changes (for example, local or remote) to export (for more information, read "ExportImportFlags Structure" on page 19).
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

_import() Method

Use this method to import changes from the log file.

NOTE: In a Java environment, you must specify the import method as _import(), which avoids conflict with the reserved word 'import'.

Signature

ImportError[] import (String importFilePath, ExportImportFlags flag, Calendar lastPurgedLog, boolean updateReplicationLog, boolean returnErrors, CatalogItemsFilter filter, String sessionID);

Argument	Description
String importFilePath	Specifies the path of the log file to import.
ExportImportFlags flag	Specifies the changes (for example, local or remote) to export (for more information, read "ExportImportFlags Structure" on page 19).
Calendar lastPurgedLog	Specifies the date and time that the log was last cleaned up. If the change in the export file was made after that time, then import uses local logs to determine if it should be replayed, otherwise it uses the last access time.
boolean updateReplicationLog	If set to TRUE, then the replication log is updated. If set to FALSE, then the replication log is not updated.
boolean returnErrors	If set to TRUE, then the function returns an array of ImportError objects which describes cases when changes recorded in the import file that satisfy filter conditions were not replayed.
CatalogItemsFilter filter	Specifies that you should filter changes made within a particular time period, and to catalog items in specified folders. Can be null.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns an ImportError structure containing the list of errors encountered. For more information, read "ImportError Structure" on page 20.

markForReplication() Method

Use this method to change the "replicable" flag on a specified folder and its descendants.

Signature

void markForReplication (String item, boolean replicate, String sessionID);

Argument	Description
String item	Specifies the path of the folder.

Argument	Description
boolean replicate	If set to TRUE, then mark the folder as replicable. If set to FALSE, then remove the replicable flag.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

ReportEditingService Service

Use this service to merge arguments and Oracle BI Web Services data to create and return the results. Table 44 shows the supported methods.

Table 44. ReportEditingService Methods

Method Names	Description
applyReportParams() Method on page 46	Applies report arguments to the report object and returns the results.
generateReportSQL() Method on page 47	Retrieves the SQL query for a given report.

applyReportParams() Method

Use this method to apply report arguments to the report and return the results.

Signature

Object applyReportParams(ReportRef object, ReportParams reportParams, boolean encodeInString, String sessionID);

Arguments	Description
ReportRef object	Specifies the path to the report definition, supplied in the ReportRef common structure. For information about the ReportRef structure, read "ReportRef Structure" on page 25.
ReportParams reportParams	Optional. Specifies the filters or variables to apply to the report before execution, supplied in the ReportParams common structure. For more information, read "Description of Services in Oracle BI Web Services" on page 33.
boolean encodeInString	If set to TRUE, then the returned report object is encoded as a character string.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns the result of applying report arguments to the specified report object. If you set encodeInString to true, then the result is encoded as a character string.

generateReportSQL() Method

Use this method to retrieve the logical SQL query for a given report.

Signature

String generateReportsSQL(ReportRef reportRef, ReportParams reportParams, String sessionID);

Arguments	Description
ReportRef reportRef	Specifies the path to the report definition supplied in the ReportRef common structure. For more information, read "ReportRef Structure" on page 25.
ReportParams reportParams	Optional. Specifies the path to the filters or variables to apply to the report before execution, supplied in the ReportParams common structure. For more information, read "Description of Services in Oracle BI Web Services" on page 33.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

A string containing the SQL query for the specified report.

SAWSessionService Service

Use this service to provide authentication methods such as logon and logoff, and other session-related methods. Table 45 shows the supported methods.

Table 45. SAWSessionService Methods

Method Name	Description
getCurUser() Method on page 48	Retrieves the current user ID for the session.
GetSessionEnvironment() Method on page 51	Retrieves the environment object for the current session?
impersonate() Method on page 48	Logs on and then impersonates the user.

Table 45. SAWSessionService Methods

Method Name	Description
impersonateex() Method on page 49	Logs on and then impersonates the user. Similar to the impersonate method, but impersonateex can specify optional session parameters.
keepAlive() Method on page 50	Instructs Oracle BI Web Services not to end particular sessions due to inactivity.
logoff() Method on page 50	Logs the user off Oracle BI Web Services.
logon() Method on page 50	Authenticates the user.
logonex() Method on page 51	Authenticates the user. Similar to the logon method, but logonex can specify optional session parameters.

getCurUser() Method

Use this method to retrieve the current user name for the session.

Signature

String getCurUser(String sessionID);

Argument	Description
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns a string indicating the current user name for the session.

impersonate() Method

Use this method to log on and impersonate the user during the SAWSessionService service. This method is useful when you need to create sessions for multiple users and have only the administrator’s name and password. You do not need to use the (logon) method if you use the impersonate() method.

If user authentication or impersonation fails, an exception is thrown.

Signature

String impersonate(String name, String password, String impersonateID);

Arguments	Description
String name	Specifies the user name to log on and authenticate.
String password	Specifies the password for the user. If there is no password for the user, leave this field empty (void).
String impersonateID	Specifies the user name to impersonate the authenticated user.

Returns

This method returns the session ID and sets an HTTP session cookie. The session ID is used in other methods to identify the Oracle BI Web Services session.

impersonateex() Method

Use this method to log on and impersonate the user in the SAWSessionService service. Similar to the impersonate method, but impersonateex can specify optional session parameters. This method is useful when you need to create sessions for multiple users and have only the administrator's name and password. You do not need to use the (logon) method if you use the impersonateex() method.

If user authentication or impersonation fails, then an exception is thrown.

Signature

AuthResult impersonateex(String name, String password, String impersonateID, SAWSessionParameters sessionparams);

Arguments	Description
String name	Specifies the user name to log on and authenticate.
String password	Specifies the password for the user. If there is no password for the user, leave this field empty (void).
String impersonateID	Specifies the user name to impersonate the authenticated user.
SAWSessionParameters ses sionparams	Optional. Specifies the session parameters to use, supplied in the SAWSessionParameters structure. For information about the SAWSessionParameters structure, read "SAWSessionParameters Structure" on page 29.

Returns

This method returns the AuthResult structure containing the session ID, and also sets an HTTP session cookie. The session ID is used in other methods to identify the Oracle BI Web Services session. For more information, see "AuthResult Structure" on page 17.

keepAlive() Method

Use this method to instruct Oracle BI Web Services not to end particular Web user sessions due to inactivity. The effect of this method on session lifetime is the same as if those users performed an activity in the browser such as clicking a report, or invoking a method. For more information about ending Web user sessions due to inactivity, read the topic “Setting the Time to Log Users Off Web Automatically” in the *Oracle Business Intelligence Presentation Services Administration Guide*.

Signature

void keepAlive(String[] sessionIDs);

Argument	Description
String[] sessionIDs	Specifies an array of session IDs to remain logged on.

logoff() Method

Use this method to log off the user from Oracle BI Web Services.

Signature

void logoff(String sessionID);

Argument	Description
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

logon() Method

Use this method to authenticate the user. If authentication fails, an exception is thrown.

Signature

String logon(String username, String password)

Arguments	Description
String username	Specifies the user name to authenticate.
String password	Specifies the password for the user. If there is no password, leave this field empty (void).

Returns

This method returns the session ID and sets an HTTP session cookie. The session ID is used in other methods to identify the Oracle BI Web Services session.

logonex() Method

Use this method to authenticate the user. Logonex() to the logon method, but logonex can specify optional session parameters. If authentication fails, an exception is thrown.

Signature

String AuthResult logonex(String username, String password,
SAWSessionParameters sessionparams);

Arguments	Description
String username	Specifies the user name to authenticate.
String password	Specifies the password for the user. If there is no password, leave this field empty (void).
SAWSessionParameters ses sionparams	Optional. Specifies the sessionparams to use, supplied in the SAWSessionParameters structure. For information about the SAWSessionParameters structure, read "SAWSessionParameters Structure" on page 29.

Returns

This method returns the AuthResult structure containing the session ID, and also sets an HTTP session cookie. The session ID is used in other methods to identify the Oracle BI Web Services session.

GetSessionEnvironment() Method

Use this method to retrieve the environment object for the current session.

Signature

GetSessionEnvironment (String sessionID);

Arguments	Description
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

This method returns a session environment object (for more information, see "SessionEnvironment Structure" on page 29).

SecurityService Service

Use this service to provide methods for identifying accounts and privileges. Table 46 shows the supported methods.

Table 46. SecurityService Methods

Method Names	Description
forgetAccount() Method on page 52	Removes an Oracle BI Web Services internal ID to account name mapping.
getCatalogAccountsDatabase() Method on page 53	Get a list of catalog accounts, filtered by user or group in the cache.
getGlobalPrivilegeACL() Method on page 53	Get the Access Control List for global privileges.
getGlobalPrivileges() Method on page 54	Get the list of all global privileges.
getPermissions() Method on page 54	Get the list of permissions for the specified user.
renameAccount() Method on page 55	Change the name of an user account.
updateGlobalPrivilegeACL() Method on page 56	Update the Access Control List for global privileges.

forgetAccount() Method

Use this method to remove an Oracle BI Web Services internal ID to account name mapping. This action is useful when an account mapping was created by mistake, for example as a side effect of an updateGlobalSAWPrivilegeACL method with a misspelled account name.

Signature

void forgetAccount(Account account, Integer cleanuplevel, String sessionID);

Argument	Description
Account account	Specifies the accounts to forget, supplied in the Account structure. For information about the Account structure, read "ACL Structure" on page 16.

Argument	Description
Integer cleanuplevel	Specifies the amount of mapping information to remove. Set to 0 to remove the mapping from an internal account ID and a user or group name. Set to 1 to remove the user directory if accounts refer to a user.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

getCatalogAccountsDatabase() Method

Use this method to retrieve a list of catalog accounts, which are filtered by user or group in the cache.

Signature

Account[] getCatalogAccountsDatabase(AccountFilter accountFilter, String sessionID)

Argument	Description
AccountFilter accountFilter	Specifies how you to filter the accounts in the cache (for information, see AccountsFilter Structure on page 16). For example, you can filter by user or group.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns an array of accounts. If accountsFilter returns a null value, this method returns all accounts cached by the current Presentation Services instance.

getGlobalPrivilegeACL() Method

Use this method to retrieve the Access Control List for global privileges.

Signature

ACL getGlobalPrivilegeACL(String privilegeName, String sessionID);

Argument	Description
String privilegeName	Specifies the name of privilege to retrieve.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns the Access Control List in an ACL structure. For more information on the ACL structure, read "ACL Structure" on page 16.

getGlobalPrivileges() Method

Use this method to retrieve the list of global privileges.

Signature

Privilege[] getGlobalPrivileges(String sessionID);

Argument	Description
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns privilege information in an array of Privileges. For more information on the Privilege structure, read "Privilege Structure" on page 22.

getPermissions() Method

Use this method to retrieve a list of permissions for the specified user, based on the specified access control list.

NOTE: This method also returns any permissions that are inherited by a user's security group, even if the access control list does not specify the group's permissions.

Signature

int getPermissions(ACL acl, Account account, String sessionID)

Argument	Description
ACL acl	Specifies the access control list for the user specified by Account account.
Account account	Specifies the user or group about which you want permissions information.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns permissions information in the permissionMask field in the AccessControlToken structure (for more information, see AccessControlToken Structure on page 15).

getCatalogAccountsDatabase() Method

Use this method to retrieve a list of catalog accounts, which are filtered by user or group in the cache.

Signature

Account[] getCatalogAccountsDatabase(AccountFilter accountFilter, String sessionID)

Argument	Description
AccountFilter accountFilter	Specifies how you to filter the accounts in the cache (for information, see AccountsFilter Structure on page 16). For example, you can filter by user or group.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns an array of accounts. If accountsFilter returns a null value, this method returns all accounts cached by the current Presentation Services instance.

renameAccount() Method

Use this method to change the name of a user account in the Presentation Catalog.

Signature

Account[] getCatalogAccountsDatabase(AccountFilter accountFilter, String sessionID)

Argument	Description
String from	Specifies the old name of the account.
String to	Specifies a new name for the account.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns an array of accounts. If accountsFilter returns a null value, this method returns all accounts cached by the current Presentation Services instance.

updateGlobalPrivilegeACL() Method

Use this method to update the Access Control List for global privileges.

Signature

void updateGlobalPrivilegeACL(String privilegeName, ACL acl,
UpdateACLParams updateACLParams, String sessionID);

Arguments	Description
String privilegeName	Specifies the name of privilege to update.
ACL acl	Specifies the Access Control List to update, supplied in the ACL structure. For information about the ACL structure, read "ACL Structure" on page 16.
UpdateACLParams updateACLParams	Specifies the Access Control List parameters to update, supplied in the UpdateACLParams structure. For information about the UpdateACLParams structure, read "UpdateACLParams Structure" on page 30.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

WebCatalogService Service

Use this service to provide methods for navigating and managing the Presentation Catalog, and to read and write Presentation Catalog objects in XML format. Table 47 shows the supported methods.

Table 47. WebCatalogService Methods

Method Names	Description
copyItem() Method on page 58	Copies an object from one location to another in the Presentation Catalog.
createFolder() Method on page 58	Creates a new folder in the Presentation Catalog.
createLink() Method on page 59	Creates a link to the Presentation Catalog.
deleteItem() Method on page 59	Deletes an object from the Presentation Catalog.
getItemInfo() Method on page 60	Retrieves Presentation Catalog information for an object.
getSubItems() Method on page 60	Retrieves the collection of child subitems for an object in the Presentation Catalog.
moveItem() Method on page 61	Moves an object in the Presentation Catalog to a different location in the catalog.
readObject() Method on page 62	Reads an object from the Presentation Catalog.
readObjects() Method on page 62	Read a list of objects from the Presentation Catalog.
removeFolder() Method on page 63	Deletes a folder from the Presentation Catalog.
setItemAttributes() Method on page 63	Sets attribute flags for the specified catalog item.
setItemProperty() Method on page 64	Sets a property for an object in the Presentation Catalog.
takeOwnership() Method on page 64	Take ownership of the specified item.
updateCatalogItemACL Method on page 65	Update the Access Control List for an item in the Presentation Catalog.
writeObject() Method on page 65	Writes an object to the Presentation Catalog.
writeObjects() Method on page 66	Writes a list of objects to the Presentation Catalog.
writeReport() Method on page 67	Writes a set of results to the Presentation Catalog.
writeDashboard() Method on page 68	Writes a dashboard object to the Presentation Catalog.
writeDashboardPrompt() Method on page 68	Writes a dashboard prompt to the Presentation Catalog.
writeDashboardPage() Method on page 69	Writes a dashboard page to the Presentation Catalog.
writeSavedFilter() Method on page 70	Writes a filter to the Presentation Catalog.

ErrorDetailsLevel Enumeration

This enumeration specifies a list of valid values for methods in the "WebCatalogService Service". Table 48 lists the values in this enumeration.

NOTE: Only one of the values in ErrorDetailsLevel should be selected.

Table 48. ErrorDetailsLevel Enumeration Values

Values	Description
String ErrorCode	Specifies that the ErrorInfo.errorCode field is populated.
String ErrorCodeAndText	Specifies that the ErrorInfo.errorCode and ErrorInfo.message fields are populated.
String FullDetails	Specifies that all ErrorInfo fields are populated.

copyItem() Method

Use this method to copy an object from one location in the Presentation Catalog to another location in the Presentation Catalog.

Signature

void copyItem(String pathSrc, String pathDest, String sessionID);

Arguments	Description
String pathSrc	Specifies the current path to the object in the Presentation Catalog.
String pathDest	Specifies the location in the Presentation Catalog where the object should be copied.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

createFolder() Method

Use this method to create a new folder in the Presentation Catalog.

Signature

void createFolder(String path, boolean createIfNotExists, String sessionID);

Arguments	Description
String path	Specifies the location in the Presentation Catalog where the folder should be created, including the name of the new folder.
boolean createIfNotExists	If set to TRUE, then the folder object is created in the Presentation Catalog if it does not already exist. If set to FALSE, then the folder object is not recreated if it already exists.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

createLink() Method

Use this method to create a link to the Presentation Catalog.

Signature

void createLink(String sPath, String sTargetPath, boolean overwriteIfExists, String sessionID);

Arguments	Description
String sPath	Specifies the path to the parent object in the Presentation Catalog.
String sTargetPath	Specifies the location in the Presentation Catalog to which the link being created should refer.
boolean overwriteIfExists	If set to TRUE, then the link is overwritten if it already exists in the Presentation Catalog. If set to FALSE, then the link is not overwritten if it already exists in the Presentation Catalog.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

deleteItem() Method

Use this method to delete an object from the Presentation Catalog. To delete a folder, read "removeFolder() Method" on page 63.

Signature

void deleteItem(String path, String sessionID);

Arguments	Description
String path	Specifies the path to the object in the Presentation Catalog.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

getItemInfo() Method

Use this method to retrieve Presentation Catalog information for an object.

Signature

ItemInfo getItemInfo(String path, boolean resolveLinks, String sessionID);

Arguments	Description
String path	Specifies the path to the object in the Presentation Catalog.
boolean resolveLinks	If set to TRUE and the path in the Presentation Catalog refers to a link, then Oracle BI Web Services retrieves information for the object pointed to by the link.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns Presentation Catalog information for an object in an ItemInfo structure. For more information, read "ItemInfo Structure" on page 21.

getSubItems() Method

Use this method to retrieve the collection of child sub-items for an object in the Presentation Catalog.

Signature

ItemInfo[] getSubItems(String path, String mask, boolean resolveLinks, GetSubItemsParams options, String sessionID);

Arguments	Description
String path	Specifies the path to the parent object in the Presentation Catalog.
String mask	Specifies a mask that indicates the child subitems to retrieve. The mask character is an asterisk (*). To retrieve all child subitems, use a single asterisk.
boolean resolveLinks	If set to TRUE and the path in the Presentation Catalog refers to a link, then information is retrieved for the child subitems of the object pointed to by the link.
GetSubItemsParams options	Optional. Specifies parameters to supply to the GetSubItemsParams structure. For information about the GetSubItemsParams structure, read "GetSubItemsParams Structure" on page 19.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns a collection of child subitems in an ItemInfo structure. For more information, read "ItemInfo Structure" on page 21.

moveItem() Method

Use this method to move an object in the Presentation Catalog to a different location in the Presentation Catalog.

Signature

void moveItem(String pathScr, String pathDest, String sessionID);

Arguments	Description
String pathSrc	Specifies the current path to the object in the Presentation Catalog.
String pathDest	Specifies the location in the Presentation Catalog where the object should be moved.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

readObject() Method

Use this method to read an object from the Presentation Catalog and return a CatalogObject structure.

Signature

CatalogObject readObject(String path, boolean resolveLinks, String sessionID);

Arguments	Description
String path	Specifies the location of the object in the Presentation Catalog.
boolean resolveLinks	If set to TRUE and the path in the Presentation Catalog refers to a link, then the object is written to the location pointed to by the link.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns a CatalogObject structure containing the specified object from the Presentation Catalog. For a description of the CatalogObject structure, read "CatalogObject Structure" on page 18.

readObjects() Method

Use this method to read a list of objects from the Presentation Catalog.

Signature

CatalogObject[] readObjects(String[] paths, boolean resolveLinks, ErrorDetailsLevel errorMode, String sessionID);

Argument	Description
String[] paths	Specifies the location of the objects in the Presentation Catalog.
boolean resolveLinks	If set to TRUE and the path in the Presentation Catalog refers to a link, then the object is written to the location pointed to by the link.
ErrorDetailsLevel errorMode	Specifies the amount of error information in the errorInfo field in the CatalogObjects structure. For more information, see CatalogObject Structure on page 18.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns an array of CatalogObjects.

NOTE: If a read operation fails for a catalog object (for example, due to an invalid path or insufficient privileges), the errorInfo field for that object contains a description of the error.

removeFolder() Method

Use this method to delete a folder and its contents from the Presentation Catalog. To delete an object other than a folder and its contents, read "deleteItem() Method" on page 59.

Signature

void removeFolder(String path, boolean recursive, String sessionID);

Arguments	Description
String path	Specifies the path to the folder in the Presentation Catalog.
boolean recursive	If set to TRUE, then remove the specified folder and its contents. If set to FALSE, then only remove the specified folder if it is empty, otherwise display an exception message.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

setItemAttributes() Method

Use this method to set attribute flags for a specified catalog item.

Signature

void setItemAttributes (String path, int attributes, String sessionID);

Arguments	Description
String path	Specifies the path to the folder in the Presentation Catalog.

Arguments	Description
int attributes	Specifies a combination of the following flags: 1 = read only 2 = archive 4 = hidden 8 = system
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

setItemProperty() Method

Use this method to set a property for an object in the Presentation Catalog.

Signature

void setItemProperty(String path, String name, String value, String sessionID);

Arguments	Description
String path	Specifies the path to the object in the Presentation Catalog.
String name	Specifies the name of the property to set.
String value	Specifies the new setting for the property.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

takeOwnership() Method

Use this method to take ownership of the specified item.

Signature

void takeOwnership(String path, boolean recursive, String sessionID);

Arguments	Description
String path	Specifies the location in the Presentation Catalog of the object to take ownership.

Arguments	Description
boolean recursive	If set to TRUE, then apply this action to the specified folder and its contents. If set to FALSE, then only apply this action to the specified folder if it is empty, otherwise display an exception message.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

updateCatalogItemACL Method

Use this method to update the Access Control List for an item in the Presentation Catalog.

Signature

void takeOwnership(String path, ACL acl, UpdateCatalogItemACLParams options, String sessionID);

Fields	Description
String path	Specifies the path to the object in the Presentation Catalog.
ACL acl	Specifies the Access Control List. For more information, see ACL Structure on page 16.
UpdateCatalogItemACLParams options	Specifies additional parameters. For more information, see UpdateCatalogItemACLParams Structure on page 31.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeObject() Method

Use this method to write an object to the Presentation Catalog in XML format.

Signature

void writeObject(CatalogObject object, String path, boolean resolveLinks, boolean allowOverwrite, String sessionID);

Arguments	Description
CatalogObject object	Specifies the object to write to the Presentation Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 18. All fields of object.itemInfo are ignored, except for the array of item properties, which are applied to the object. The signature of the resulting document is always COXmlDocument1.
String path	Specifies the location in the Presentation Catalog where the object should be written.
boolean resolveLinks	If set to TRUE and the path in the Presentation Catalog refers to a link, then the object is written to the location pointed to by the link.
boolean allowOverwrite	If set to TRUE, then if the object already exists in the Presentation Catalog, it is overwritten. If set to FALSE, then if the object already exists in the Presentation Catalog, it is not overwritten.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeObjects() Method

Use this method to write an array of objects to the Oracle BI Web Services Presentation Catalog.

Signature

ErrorInfo[] writeObjects(CatalogObject[] catalogObjects, boolean allowOverwrite, ErrorDetailsLevel errorMode, String sessionID);

Argument	Description
CatalogObject[] object	Specifies the objects to write to the Presentation Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, see "CatalogObject Structure" on page 18. All fields of object.itemInfo are ignored, except for the array of item properties, which are applied to the object. The signature of the resulting document is always COXmlDocument1.
boolean allowOverwrite	If set to TRUE, then if the object already exists in the Presentation Catalog, it is overwritten. If set to FALSE, then if the object already exists in the Presentation Catalog, it is not overwritten.

Argument	Description
ErrorDetailsLevel errorMode	Specifies the amount of error information in the errorInfo field in the CatalogObjects structure.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

An array of ErrorInfo objects.

writeReport() Method

Use this method to write a set of results to the Presentation Catalog.

Signature

void writeReport(CatalogObject object, String path, boolean resolveLinks, boolean allowOverwrite, String sessionID);

Arguments	Description
CatalogObject object	Specifies the object to write to the Presentation Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 18. All fields of object.itemInfo are ignored, except for the array of item properties, which are applied to the object. The signature of the resulting document is always queryitem1.
String path	Specifies the location in the Presentation Catalog where the results should be written.
boolean resolveLinks	If set to TRUE and the path in the Presentation Catalog refers to a link, then the results are written to the location pointed to by the link.
boolean allowOverwrite	If set to TRUE, then if the results already exist in the Presentation Catalog, they are overwritten. If set to FALSE, then if the results already exist in the Presentation Catalog, they are not overwritten.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeDashboard() Method

Use this method to write a dashboard object to the Presentation Catalog.

Signature

void writeDashboard(CatalogObject object, String path, boolean resolveLinks,
boolean allowOverwrite, String sessionID);

Arguments	Description
CatalogObject object	Specifies the dashboard object to write to the Presentation Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 18. All fields of object.itemInfo are ignored, except for the array of item properties, which are applied to the object. The signature of the resulting document is always dashboarditem1.
String path	Specifies the location in the Presentation Catalog where the dashboard object should be written.
boolean resolveLinks	If set to TRUE and the path in the Presentation Catalog refers to a link, then the dashboard object is written to the location pointed to by the link.
boolean allowOverwrite	If set to TRUE, then if the dashboard object already exists in the Presentation Catalog, it is overwritten. If set to FALSE, then if the dashboard object already exists in the Presentation Catalog, it is not overwritten.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeDashboardPrompt() Method

Use this method to write a dashboard prompt to the Presentation Catalog.

Signature

void writeDashboardPrompt(CatalogObject object, String path, boolean resolveLinks, boolean allowOverwrite, String sessionID);

Arguments	Description
CatalogObject object	Specifies the dashboard prompt object to write to the Presentation Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 18. All fields of object.itemInfo are ignored, except for the array of item properties, which are applied to the object. The signature of the resulting document is always globalfilteritem1.
String path	Specifies the location in the Presentation Catalog where the dashboard prompt should be written.
boolean resolveLinks	If set to TRUE and the path in the Presentation Catalog refers to a link, then the dashboard prompt is written to the location pointed to by the link.
boolean allowOverwrite	If set to TRUE, then if the dashboard prompt already exists in the Presentation Catalog, it is overwritten. If set to FALSE, then if the dashboard prompt already exists in the Presentation Catalog, it is not overwritten.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeDashboardPage() Method

Use this method to write a dashboard page to the Presentation Catalog.

Signature

void writeDashboardPage(CatalogObject object, String path, boolean resolveLinks, boolean allowOverwrite, String sessionID);

Arguments	Description
CatalogObject object	Specifies the dashboard page object to write to the Presentation Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 18. All fields of object.itemInfo are ignored, except for the array of item properties, which are applied to the object. The signature of the resulting document is always dashboardpageitem1.
String path	Specifies the location in the Presentation Catalog where the dashboard page should be written.
boolean resolveLinks	If set to TRUE and the path in the Presentation Catalog refers to a link, then the dashboard page is written to the location pointed to by the link.
boolean allowOverwrite	If set to TRUE, then if the dashboard page already exists in the Presentation Catalog, it will be overwritten. If set to FALSE, then if the dashboard page already exists in the Presentation Catalog, it will not be overwritten.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeSavedFilter() Method

Use this method to write a filter to the Presentation Catalog.

Signature

void writeSavedFilter(CatalogObject object, String path, boolean resolveLinks, boolean allowOverwrite, String sessionID);

Arguments	Description
CatalogObject object	Specifies the filter object to write to the Presentation Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 18. All fields of object.itemInfo are ignored, except for the array of item properties, which are applied to the object. The signature of the resulting document is always savedfilteritem1.
String path	Specifies the location in the Presentation Catalog where the filter should be written.
boolean resolveLinks	If set to TRUE and the path in the Presentation Catalog refers to a link, then the filter is written to the location pointed to by the link.
boolean allowOverwrite	If set to TRUE, then if the filter already exists in the Presentation Catalog, it is overwritten. If set to FALSE, then if the filter already exists in the Presentation Catalog, it is not overwritten.
String sessionID	Specifies the session ID, which is usually returned by the logon method. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

XMLViewService Service

Use this service to retrieve results from Oracle BI Web Services in XML format. Table 49 shows the supported methods.

Table 49. XMLView Service Methods

Method Name	Description
"cancelQuery() Method"	Cancels the current query.
"executeSQLQuery() Method"	Runs a SQL query.
"executeXMLQuery() Method"	Runs an XML query.
"fetchNext() Method"	Returns the next page of data rows.

XMLQueryOutputFormat Enumeration

This enumeration specifies a list of valid values for the executeSQLQuery() Method and executeXMLQuery() Method. For example, you might want to return data rows and metadata, or data rows only. Table 50 lists the values in this enumeration.

NOTE: Only one of the values in XMLQueryOutputFormat can be selected.

Table 50. XMLQueryOutputFormat Enumeration Values

Values	Description
String SAWRowsetData	Specifies that the query returns only data rows.
String SAWRowsetSchema	Specifies that the query returns only metadata.
String SAWRowsetSchemaAndData	Specifies that the query returns both metadata and data rows.

cancelQuery() Method

Use this method to cancel a query and clean up resources associated with the query. This method should only be used if the query row set is not scrolled to the last row in the data set returned.

NOTE: If you use this method when the query row set is scrolled to the last row in the data set returned, query data is cleaned up during the last fetchNext method invocation.

Signature

QueryResults cancelQuery(String queryID, String sessionID);

Argument	Description
String queryID	Specifies the unique ID of the query.
String sessionID	Specifies the unique ID of the session.

executeSQLQuery() Method

Use this method to execute a SQL query and return the results of the query.

NOTE: If the results returned exceed one page, you need to use the fetchNext() Method to return the next page of rows.

Signature

QueryResults executeSQLQuery(String sql, XMLQueryOutputFormat outputFormat, XMLQueryExecutionOptions executionOptions, String sessionID);

Argument	Description
String sql	Specifies the string of SQL code to execute.
XMLQueryOutputFormat out putFormat	Specifies the output format (for more information, see on page 32).
XMLQueryExecutionOptions executionOptions	Specifies the query execution options (for more information, see XMLQueryExecutionOptions Structure on page 32).
String sessionID	Specifies the unique ID of the session.

Returns

Returns the results of the query as one or more rows of data in a QueryResults structure (for more information, see QueryResults Structure on page 22).

executeXMLQuery() Method

Use this method to execute an XML query and return the results of the query.

NOTE: If the results returned exceed one page, you need to use the fetchNext() Method to return the next page of rows.

Signature

QueryResults executeXMLQuery(ReportRef report, XMLQueryOutputFormat outputFormat, XMLQueryExecutionOptions executionOptions, ReportParams reportParams, String sessionID);

Argument	Description
ReportRef reportRef	Specifies the report definition, supplied in the ReportRef common structure. For more information, read “ReportRef Structure” on page 25.
XMLQueryOutputFormat out putFormat	Specifies the output format (for more information, see on page 32).
XMLQueryExecutionOptions executionOptions	Specifies the query execution options (for more information, see XMLQueryExecutionOptions Structure on page 32).

Argument	Description
ReportParams reportParams	Optional. Specifies the filters or variables to apply to the report before execution, supplied in the ReportParams common structure. For information about the ReportParams structure, read "ReportParams Structure" on page 24.
String sessionID	Specifies the unique ID of the session.

Returns

Returns the results of the query as one or more rows of data in a QueryResults structure (for more information, see QueryResults Structure on page 22).

fetchNext() Method

Use this method to return the next page of rows retrieved by a query.

NOTE: The page returned might contain zero rows. If the finished flag is not set, the remaining rows might not be available immediately.

Signature

QueryResults fetchNext(String queryID, String sessionID);

Argument	Description
String queryID	Specifies the unique ID of the query, which is returned in the QueryResults object.
String sessionID	Specifies the unique ID of the session.

Returns

Returns the next page of query results as one or more rows of data in a QueryResults structure (for more information, see QueryResults Structure on page 22).

5 Format of Returned Recordsets

This is the basic structure for Oracle BI Web Services rowset XML output:

```
<Recordset xmls="OracleBI NS" >

     <xsd::schema xsd = ...>
          .
          .
          .
     <xsd::schema>

     <row>

            <column1>value1</column1>

            <column2>value2</column2>

     </row>

     <row>...</row>

     <row>...</row>

</Recordset>
```

Each row element holds the contents of one SQL record. Child elements of the row contain values of record fields. The recordset XML may optionally include XSD schema that describe the format of row elements.

6 Code Example

The following C# code example uses Oracle BI Web Services to extract Presentation Catalog information and write it to XML files.

NOTE: Some code that appears on a single line in an application development environment may appear on more than one line when the code is printed or viewed online because of page or window size limitations.

```
using System;
using System.IO;
using System.Web;

using CatalogExport.SAWServices;

namespace CatalogExport
{
   /// <summary>
   /// Summary description for Class1.
   /// </summary>
   class CatalogExport
   {
      static private System.Net.CookieContainer cookies = new
System.Net.CookieContainer ();
      static private SAWSessionService m_session = new SAWSessionService ();
      static private WebCatalogService m_WebCatalogService = new WebCatalogService();
      static int m_nCurFileIndex=0;
      static StreamWriter m_curFile = null;
      static int m_nFileMaxlen=1024*1024*5;
      static String m_strExportDir=null;
      static String m_strFilePrefix="catalog";

      static void openFile()
      {
         if (m_curFile== null || m_curFile.BaseStream.Length > m_nFileMaxlen)
         {
            if (m_curFile != null )
            {
               m_curFile.WriteLine("</CatalogRoot>");
               m_curFile.Close();
            }
            String strNewPath = m_strExportDir + "\\" + m_strFilePrefix +
(++m_nCurFileIndex) + ".xml";
            m_curFile = new StreamWriter(strNewPath);
            m_curFile.WriteLine("<CatalogRoot>");
         }
      }
      /// <summary>
      /// The main entry point for the application.
      /// </summary>
```

```
[STAThread]
static void Main(String[] args)
{
   String strURL="http://localhost/analytics/saw.dll";
   String strUser="Administrator";

   String strPWD="";

   for (int i=0;i<args.Length;++i)
   {
      if (args[i].Equals("/URL"))
         strURL = args[++i];
      else if (args[i].Equals("/USER"))
         strUser =  args[++i];
      else if (args[i].Equals("/PWD"))
         strPWD =  args[++i];
      else if (args[i].Equals("/DIR"))
         m_strExportDir =  args[++i];
      else if (args[i].Equals("/?"))
      {
         printUsage();
         return;
      }
   }

   if (m_strExportDir == null)
   {
      printUsage();
      return;
   }
   Directory.CreateDirectory(m_strExportDir);
   //let all services  use the same cookie container - so all of them
   //would have access to Session cookie
   m_WebCatalogService.CookieContainer  = cookies;
   m_session.CookieContainer  = cookies;
   m_session.Url = strURL + "?SoapImpl=nQSessionService";
   m_WebCatalogService.Url = strURL + "?SoapImpl=webCatalogService";
   String sessionID = m_session.logon(strUser,strPWD);
   try
   {
      processCatalogFolder("/",sessionID);
   }
   finally
   {
      if (m_curFile != null )
      {
         m_curFile.WriteLine("</CatalogRoot>");
         m_curFile.Close();
      }
   }

}

static void processCatalogFolder(String path,String sessionID)
```

```
      {
         ItemInfo[] arrChilds =
m_WebCatalogService.getSubItems(path,"*",false,null,sessionID);
         foreach (ItemInfo info in arrChilds)
         {
            switch (info.type)
            {
               case ItemInfoType.Folder:
                  try
                  {
                     processCatalogFolder(info.path,sessionID);
                  }
                  catch (Exception e)
                  {
                     Console.WriteLine(e.Message);
                  }

                  continue;
               case ItemInfoType.Object:
               {
                  if (!isKnownSignature(info.signature))
                     continue;
                  openFile();
                  CatalogObject co =
m_WebCatalogService.readObject(info.path,true,sessionID);
                  m_curFile.WriteLine("<CatalogObj path=\"" +
HttpUtility.HtmlEncode(info.path) + "\" signature=\"" + info.signature +"\">");
                  m_curFile.WriteLine(co.catalogObject.ToString());
                  m_curFile.WriteLine("</CatalogObj>");
                  break;
               }
            }
         }

      }

      static bool isKnownSignature(String strSignature)
      {
         return strSignature=="dashboardpageitem1" ||
               strSignature=="dashboarditem1" ||
               strSignature=="queryitem1"  ||
               strSignature=="dashboarditem1"  ||
         strSignature=="globalfilteritem1"  ||
         strSignature=="filteritem1" ||
         strSignature=="COXmlDocument1";
      }
      static void printUsage()
      {
         Console.WriteLine("CatalogExport /DIR exportdir [/USER usename] [/PWD
password] [/URL serverurl]");
      }sw
   }
}
```

Index