Siebel Analytics Web Services Guide

Version 7.8.4

December 2005

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of the Siebel Analytics Web SOAP API

Chapter 3: Description of Structures in the Siebel Analytics Web SOAP API

Chapter 4: Description of Siebel Analytics Web SOAP API Methods

Chapter 5: Format of Returned Recordsets

Chapter 6: Code Example

Index

1 What's New in This Release

What's New in Siebel Analytics Web Services Guide, Version 7.8.4

Table 1 lists changes described in this version of the documentation to support release 7.8.4 of the software.

Table 1. New Product Features in Siebel Analytics Web Services Guide, Version 7.8.4

Topic	Description
Entire Book	Removed all SOAP API information from the *Siebel Analytics Web Administration Guide* and created this *Siebel Analytics Web Services Guide*.
getResults() Method on page 59	Revised definitions of the outputFormat field.
setItemAttributes() Method on page 54	Added new setItemAttributes method.
SOAP Licensing and Permissions on page 11	Added new section on licensing and permissions.
UpdateACLParams Structure on page 26	Revised definitions of the updateFlag field.

2 Overview of the Siebel Analytics Web SOAP API

This guide describes the Siebel Analytics Web implementation of the Simple Object Access Protocol (SOAP) application programming interface to extract and deliver data and manage content in the Siebel Analytics Web Catalog.

This chapter contains the following sections:

Using the SOAP API

SOAP (Simple Object Access Protocol) is a World Wide Web Consortium (W3C) recommendation for an XML protocol for exchanging information on the Web. The Siebel Analytics Web implementation of SOAP allows you to perform two types of functions:

- Extract results from Siebel Analytics Web and deliver them to external applications.
- Perform some Web Catalog management functions.

The Siebel Analytics Web SOAP API allows external applications such as J2EE and .NET to use Siebel Analytics as an analytical calculation and data integration engine. It provides a set of Web services that allow external applications to communicate with the Siebel Analytics Web server. You can use the Siebel Analytics Web SOAP API to extract results from Siebel Analytics Web and deliver them to external applications and Web application environments. You can reference a saved report or send the criteria for the report to Siebel Analytics Web.

The formal definition of Siebel Analytics Web SOAP services can be retrieved in WSDL (Web Services Definition Language) format. Proxy classes for the services are generated automatically.

The XML Schema Definition (XSD) file for the services is the file SawServices.xsd is located in the \Web\App\Wsdl\Schema directory in the Siebel Analytics installation directory. The XSD file is used internally and cannot be used separately. You can access the WSDL document through the following Siebel Analytics Web URL:

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

The Siebel Analytics Web SOAP API has been tested with 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 the Siebel Analytics Web SOAP API

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 Siebel Analytics Web SOAP Services Interface

You can access the Siebel Analytics Web SOAP services interface on any platform on which a SOAP
client library and tools are available. The steps to access the SOAP services depend on your
programming environment.

Example of Accessing the Siebel Analytics Web SOAP API from Microsoft Visual Studio

The following procedure provides the steps to access the SOAP services from Microsoft Visual Studio.

To access the SOAP 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 appears.

3   In the URL field, type the URL to access the Siebel Analytics Web WSDL document.

The following URL is an example URL to access the Siebel Analytics Web Server WSDL document:

http://localhost/analytics/saw.dll?WSDL

4   Click Go.

The found services and methods appear in the Add Web Reference dialog box window.

5   Click the Add Reference button.

The Add Web Reference dialog box closes, and the node that represents the added Web reference appears 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 appear in the Object Browser window.

7   Begin using the classes in your program.

For a code example, see .

SOAP Licensing and Permissions

The SOAP API is available to licensed users of Siebel Analytics. 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 a SOAP call, 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 Siebel Analytics throws an "Access Denied" exception for SOAP calls that require authentication (examples: logon and logonex).

3 Description of Structures in the Siebel Analytics Web SOAP API

This chapter describes the structures used in the Siebel Analytics Web SOAP API.

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:

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	"Expression Structure" on page 16
	"ReportParams Structure" on page 20
	"ReportRef Structure" on page 21
	"Variable Structure" on page 26
HtmlViewService	"ReportHTMLOptions Structure" on page 20
	"StartPageParams Structure" on page 25
MetadataService	"SAColumn Structure" on page 22
	"SASubjectArea Structure" on page 24
	"SATable Structure" on page 24
ReplicationService	"CatalogItemsFilter Structure" on page 16
	"ImportError Structure" on page 17
SAWSessionService	"SAWLocale Structure" on page 24
	"SAWSessionParameters Structure" on page 25
SecurityService	"AccessControlToken Structure" on page 15
	"Account Structure" on page 15
	"ACL Structure" on page 15
	"Privilege Structure" on page 19
	"UpdateACLParams Structure" on page 26
WebCatalogService	"CatalogObject Structure" on page 16
	"GetSubItemsParams Structure" on page 17
	"ItemInfo Structure" on page 18
	"NameValuePair Structure" on page 19

AccessControlToken Structure

The AccessControlToken structure describes permissions granted to a specific account in the access control list. This structure is used in the Security service. Table 3 lists the fields in this structure.

Table 3.   AccessControlToken Structure Fields

Fields	Description
Account account	Reference to Account structure.
int permissionMask	A combination of the following flags: 1 = Permission to read items content 2 = Permission to traverse directory 4 = Permission to change items' content 8 = Permission to delete an item 16 = Permission to assign permissions to others 32 = Can take ownership of the item

Account Structure

The Account structure holds user or group names. It has a flag to indicate if the name is user or group. This structure is used in the Security service. Table 4 lists the fields in this structure.

Table 4.   Account Structure Fields

Fields	Description
String accountName	String to hold an account name.
int accountType	Flag where 0 indicates the account is a user, and 1 indicates a group.

ACL Structure

The ACL structure holds the access control list. This structure is used in the Security service. Table 5 lists the fields in this structure.

Table 5.   ACL Structure Fields

Fields	Description
AccessControlToken[] accessControlTokens	Full list of permissions.
Account owner	Owner of the resource.

CatalogItemsFilter Structure

The CatalogItemsFilter structure filters catalog items and changes based on the path and timestamp. This structure is used in the ReplicationService service. Table 6 lists the fields in this structure.

Table 6.   CatalogItemsFilter Structure Fields

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

CatalogObject Structure

The CatalogObject structure retrieves or specifies all information for a particular Web Catalog object in a single call. This structure is used in the WebCatalogService service. Table 7 lists the fields in this structure.

Table 7.   CatalogObject Structure Fields

Fields	Description
ItemInfo itemInfo	Web Catalog information about the object, supplied in the ItemInfo common structure.
	For information about the ItemInfo structure, read "ItemInfo Structure" on page 18.
Object catalogObject	A character string that contains an XML representation of the object, or the object itself.
	When this field is the object itself, the SOAP engine you are using should be able to provide automatic serialization to and deserialization from the XML.

Expression Structure

The Expression structure performs operations on columns in results returned in XML format. This structure holds arithmetic and logical expressions in results. Each expression operates on a column in the results and produces one of the standard SQL data types as the result, which are: boolean, DateTime, int, Object, String, and void.

The Expression structure is common to all services in the Siebel Analytics Web SOAP API. Expression() is an object reference and has no members.

GetSubItemsParams Structure

The GetSubItemsParams structure contains optional parameters used in a getSubItems call. This structure is used in the WebCatalogService service. Table 8 lists the fields in this structure.

Table 8.   GetSubItemsParams Structure Fields

Fields	Descriptions
filter	Used for internal purposes only. This field should be null.
includeACL	If true, ACL information is included in the resulting ItemInfo structures.
withPermission withPermissionMask	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.
withAttributes withAttributesMask	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.

ImportError Structure

The ImportError structure describes the cause of failure to replay particular change during import. This structure is used in the ReplicationService service. Table 9 lists the fields in this structure.

Table 9.   ImportError Structure Fields

Fields	Description
String item	The path to the changed item.

Table 9.   ImportError Structure Fields

Fields	Description
String operation	For internal use only.
String file	
int line	
String catalogError	An error string, describing the reason for the failure.

ItemInfo Structure

The ItemInfo structure contains Web Catalog information about an object. This structure is used in the WebCatalogService service. Table 10 lists the fields in this structure.

Table 10.   ItemInfo Structure Fields

Fields	Description
String path	The path to the object in the Web Catalog.
String type	A character string that indicates the type. Valid values are: ■ Folder ■ Link ■ Missing ■ NoAccess ■ Object
DateTime lastModified	The date and time that the object was last modified, in DateTime format.
DateTime created	The date and time that the object was created (saved) in the Web Catalog, in DateTime format.
DateTime accessed	The data and time that the object was last accessed by a user, in DateTime format.
String signature	The signature of the Web Catalog object. For more information about signatures, read "Using Item Signatures in the Siebel Analytics Web SOAP API" on page 10.
NameValuePair[] itemProperties	An array of object properties.

Table 10.   ItemInfo Structure Fields

Fields	Description
ACL acl	Access Control List for this catalog item.
int attributes	Combination of the following flags: 1 = read only 2 = archive 4 = hidden 8 = system

NameValuePair Structure

The NameValuePair structure denotes named properties such as COLOR=RED. This structure is used in the WebCatalogService service. Table 11 lists the fields in this structure.

Table 11.   NameValuePair Structure Fields

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

Privilege Structure

The Privilege structure represents global privileges. In Siebel Analytics Web user interface you configure these privileges using the Manage Privileges screen. This structure is used in the Security service. Table 12 lists the fields in this structure.

Table 12.   Privilege Structure Fields

Fields	Description
String name	String that contains the name of a privilege.
String description	String that contains a description of the privilege.

ReportHTMLOptions Structure

The ReportHTMLOptions structure defines options for displaying results on an HTML page. This structure is used in the HtmlViewService service. For information about the HtmlViewService service, read "HtmlViewService Service" on page 27. Table 13 lists the field in this structure.

Table 13.   ReportHTMLOptions Structure Field

Field	Description
boolean enableDelayLoading	A boolean value, 1 (true) or 0 (false).
	When set to true, the Siebel Analytics Web server is not required to provide results immediately, and may display a message indicating that it is waiting for results.
	NOTE: The value is always assumed to be true, even when set to false.
	Using enableDelayLoading can be useful when results may take some time to obtain but you want to provide immediate feedback.

ReportParams Structure

The ReportParams structure replaces existing filters and variables in a report. This structure is common to all services in the Siebel Analytics Web SOAP API. Table 14 lists the fields in this structure.

Table 14.   ReportParams Structure Fields

Fields	Description
Object[] filterExpressions	An array of Siebel Analytics Web filter expressions or their character string representations, in the form Object[] filter_expression, filter_expression …
Variable [] variables	An array of Siebel Analytics Web variables represented as character strings, in the form Variable [] variable, variable …
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 15 shows how filter expressions are applied to a report.

Table 15.   How Filter Expressions Are Applied to a Report in the Siebel Analytics Web SOAP API

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 16 shows how variables are applied to a report.

Table 16.   How Variables Are Applied to a Report in the Siebel Analytics Web SOAP API

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

The ReportRef structure references 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 the ReportRef fields should be populated.

The ReportRef structure is common to all services in the Siebel Analytics Web SOAP API. Table 17 lists the fields in this structure.

Table 17.   ReportRef Structure Fields

Fields	Description
String reportPath	A string value that provides the path to the report in the Web Catalog.
ReportDef reportDef	Should be set to NULL.
String reportXML	A string value that contains the XML that defines the report.

SAColumn Structure

The SAColumn structure represents the logical column in the Subject Area. This structure is used in the MetadataService. Table 18 lists the fields in this structure.

Table 18.   SAColumn Structure Fields

Fields	Description
String name	Column name used in SQL statements.
String displayName	Localized name, used in the Answers screen.
String description	A string to contain the description of the column name.
boolean nullable	Flag to indicate if the column is nullable or not.
SADataType dataType	Indicates the type of data a column contains. For more information, read SADataType Values on page 22.
boolean aggregateable	Flag to indicate if the column can be aggregated or not.
AggregationRule aggrRule	If the column contains aggregated data, this value indicates the type of aggregation used. For more information, read AggregationRule Values on page 23.

SADataType Values

The SADataType indicates the type of data 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 *Siebel Analytics Web 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

SATable Structure

The SATable structure represents the logical table in the Subject Area. This structure is used in the MetadataService. Table 19 lists the fields in this structure.

Table 19.   SATable Structure Fields

Fields	Description
String name	Table name used in SQL statements.
String displayName	Localized name, used in the Answers screen.
String description	A string to contain the description of the table name.
SAColumn columns	Collection of this table's columns. For information about the SAColumn structure, read "SAColumn Structure" on page 22.

SASubjectArea Structure

The SASubjectArea structure represents Subject Area attributes. This structure is used in the MetadataService. Table 20 lists the fields in this structure.

Table 20.   SASubjectArea Structure Fields

Fields	Description
String name	Table name used in SQL statements.
String displayName	Localized name, used in the Answers screen.
String description	A string to contain the description of the subject area.
SATable tables	Collection of tables for this subject area. For information about the SATable structure, read "SATable Structure" on page 24.

SAWLocale Structure

The SAWLocale structure defines the locale for the current session. This structure is used in the SAWSessionService. Table 21 lists the fields in this structure.

Table 21.   SAWLocale Structure Fields

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

SAWSessionParameters Structure

The SAWSessionParameters structure defines the optional parameters for the current session. This structure is used in the SAWSessionService. Table 22 lists the fields in this structure.

Table 22.   SAWSessionParameters Structure Fields

Fields	Description
SAWLocale locale	The locale to be used, supplied in the SAWLocale structure. For information about the SAWLocale structure, read "SAWLocale Structure" on page 24.
String userAgent	Set this field if the HTMLView service will be used with current session. It specifies the userAgent string of the browser, where Siebel Analytics Web Server HTML content is displayed. Siebel Analytics Web Server uses this information to produce browser-specific HTML.
String features	For internal use only. Should be null.

StartPageParams Structure

The StartPageParams structure is used in startPage calls. This structure is used in the HTMLView service. Table 23 lists the fields in this structure.

Table 23.   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.
String dontUseHttpCookies	Flag. If TRUE, then Siebel Analytics Web Server can not rely on cookies for passing the sessionID. Instead, the sessionID is included as a parameter in callback URLs.

UpdateACLParams Structure

The UpdateACLParams structure is used in updateACL calls. This structure is used in the Security
service. Table 24 lists the fields in this structure.

Table 24.   UpdateACLParams Structure Fields

Fields	Description
boolean bAllowUnknownAccounts	Flag. If set to TRUE(1), and the new access control list (ACL) in the updateACL includes accounts that are unknown to Siebel Analytics Web, then Siebel Analytics Web creates new account records for them. However, to be used such accounts should exist in Siebel Analytics Server as well.
int updateFlag	Flag that indicates the mode for UpdateAcl. 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.

Variable Structure

The Variable structure references a variable in the report and replaces it with another variable. This
structure is common to all services in the Siebel Analytics Web SOAP API. Table 25 lists the fields in
this structure.

Table 25.   Variable Structure Fields

Fields	Description
String name	A character string that contains the name of the variable to replace.
String valueXml	The data type of the new variable, such as String, int, or boolean, or an Expression structure.
String value	The XML representation of the new variable, encoded as a character string. **NOTE:** If name or valueXml is populated, this field must be null.

4 Description of Siebel Analytics Web SOAP API Methods

This chapter describes the methods in the Siebel Analytics Web SOAP API. Methods are described using JavaScript-like syntax and standard types. The exact signature of generated classes and methods depends on the SOAP code generation tool and the target language used by your application development environment.

This chapter covers the methods available for the following services:

HtmlViewService Service

The HtmlViewService service embeds Siebel Analytics Web 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 Siebel Analytics Web content with the content of third-party Web pages.

HTML methods extract fragments of HTML code that can be inserted in third-party Web pages. Table 26 describes the HTML code excerpts and desired page locations.

Table 26.   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 Siebel Analytics Web, the interface displays the Siebel Analytics Web "Searching…" image (the default image is a spinning arrow) embedded on the third-party Web page.

For smooth report transitioning, Siebel Analytics Web tracks the Analytics reports that have been added to a third-party Web page 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 Siebel Analytics Web server. Although it is possible to route requests directly to the Siebel Analytics Web server, in many cases it is preferable to route requests through the Web server that originally serviced the third-party page. Also, in situations where Siebel Analytics Web 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 reroute requests to Siebel Analytics Web 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 32.

Table 27 shows the supported methods for the HtmlViewService.

Table 27.   HtmlViewService Methods

Method Name	Description
addReportToPage() Method on page 29	Adds results to an HTML page.
endPage() Method on page 29	Destroys a server page object and all data associated with it.
getCommonBodyHTML() Method on page 30	Gets HTML to include in the <BODY> section.
getHeadersHTML() Method on page 30	Gets HTML to include in the <HEAD> section.
getHTMLForReport() Method on page 31	Gets HTML to display a particular set of results.
setBridge() Method on page 32	Specifies a bridge URL to receive communications. Can be useful when the Siebel Analytics Web server and the user's Web server reside on different machines or when you want to modify the results in your application development environment.
startPage() Method on page 33	Creates a new page object and returns its ID.

addReportToPage() Method

The addReportToPage() method adds 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	A character string page ID returned by the startPage() method. For information about the startPage () method, read "startPage() Method" on page 33.
String reportID	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 SOAP calls; for example, corresponding user interface elements generated by the Siebel Analytics Web server would reference the same ID.
ReportRef report	The report definition, supplied in the ReportRef structure. For more information, read "ReportRef Structure" on page 21.
String reportViewName	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. The filters or variables to apply to the report before execution, supplied in the ReportParams common structure. For more information, read "Description of Siebel Analytics Web SOAP API Methods" on page 27.
ReportHTMLOptions options	Optional. The display options to apply to the report after execution, supplied in the ReportHTMLOptions structure. For more information, read "ReportHTMLOptions Structure" on page 20.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

endPage() Method

The endPage() method destroys the Siebel Analytics Web server page object and all data associated with it.

Signature

void endpage(String pageID, String sessionID);

Arguments	Description
String pageID	A character string that contains the name of the page.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

getCommonBodyHTML() Method

The getCommonBodyHTML() method gets HTML to include in the <BODY> section.

Signature

String getCommonBodyHTML(String pageID, String sessionID);

Arguments	Description
String pageID	A character string page ID returned by the startPage() method. For information about the startPage () method, read "startPage() Method" on page 33
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. 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

The getHeadersHTML() method gets HTML to include in the <HEAD> section.

Signature

String getHeadersHTML(String pageID, String sessionID);

Arguments	Description
String pageID	A character string page ID returned by the startPage() method. For information about the startPage () method, read "startPage() Method" on page 33.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. 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

The getHTMLForReport() method gets an HTML excerpt to display the results for a particular report. Before issuing this call, 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	A character string page ID returned by the startPage() method. For information about the startPage () method, read "startPage() Method" on page 33.
String pageReportID	A character string ID returned by the addReportToPage() method.

For information about the addReportToPage method, read "addReportToPage() Method" on page 29. |
| String sessionID | A character string that identifies the session ID. The session ID is usually returned by the logon call. 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

The setBridge() method specifies a bridge URL to receive communications. Specifying a bridge URL can be useful when the Siebel Analytics Web 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 Siebel Analytics Web server are sent to the bridge URL, which then forwards requests to the Siebel Analytics Web server.

Signature

void setBridge(String bridgeURL, String sessionID);

Arguments	Description
String bridgeURL	The bridge URL.
String sessionID	A character string that contains the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Usage

You are responsible to 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 Siebel Analytics Web resource in the RedirectURL argument of the request character string. For information about the RedirectURL argument, read "How Callback URLs Are Replaced" on page 32.

■ Forward all other request arguments, together with all headers and the request body, to the bridge URL.

■ Copy the response from the Siebel Analytics Web 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, Siebel Analytics Web 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

The startPage() method creates a new page object and returns its ID.

Signature

String startPage(StartPageParams options, String sessionID);

Arguments	Description
StartPageParams options	The options to use when starting the page, supplied in the StartPageParams structure. For information about the StartPageParams structure, read "StartPageParams Structure" on page 25.
String sessionID	A character string that contains the session ID. The session ID is usually returned by the logon call. 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 Siebel Analytics Web Server page ID.

MetadataService Service

Use the MetadataService service to retrieve descriptions of Siebel Analytics schema objects: columns, tables, and subject areas. Table 28 shows the supported methods.

Table 28.   MetadataService Methods

Method Names	Description
describeColumn() Method on page 33	Retrieves column information for a specified column in a specified subject area and table.
describeSubjectArea() Method on page 34	Retrieves subject area information for a specified subject area.
describeTable() Method on page 35	Retrieves table information for a specified table in a specified subject area.
getSubjectAreas() Method on page 36	Retrieves the list of subject areas available.

describeColumn() Method

Retrieves 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	String to specify the subject area to be queried.
String tableName	String to specify the table to be queried.
String columnName	String to specify the column to be queried.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. 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 22.

describeSubjectArea() Method

Retrieves subject area information for a specified subject area.

Signature

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

Arguments	Description
String subjectAreaName	String to specify the subject area to be queried.
SASubjectAreaDetails detailsLevel	Specifies what information should be retrieved about the subject area. For information on the SASubjectAreaDetails structure, read "SASubjectAreaDetails Values" on page 35.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

SASubjectAreaDetails Values

Used to specify what information should be retrieved about the subject area. Table 29 lists the available values.

Table 29.   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 information on the SASubjectArea structure, read "SASubjectArea Structure" on page 24.

Usage

Depending on the value of the detailsLevel parameter, the returned Object contains the information specified in Table 30.

Table 30.   Contents of SASubjectArea Object Based on detailsLevel Parameter

Value of detailsLevel	Description
IncludeTables	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	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	Table list is not available. The tables field in the resulting subject area object is null.

describeTable() Method

Retrieves 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	String to specify the subject area to be queried.
String tableName	String to specify the table to be queried.
SATableDetails detailsLevel	Specifies what information should be retrieved about the table. For information on the SATableDetails structure, read "SATablesDetails Values" on page 36.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. 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 what information should be retrieved about the table. Table 31 lists the available values.

Table 31.   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 24.

getSubjectAreas() Method

Retrieves the list of subject areas available.

Signature

SASubjectArea getSubjectAreas(String sessionID);

Arguments	Description
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns an SASubjectArea Object. For information on the SASubjectArea structure, read "SASubjectArea Structure" on page 24.

Usage

SASubjectArea objects returned by this call 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 get the list of tables. Then use describeTable() to retrieve the list of columns in a specified table, and finally use describeColumn() to get information on a specified column.

ReplicationService Service

The ReplicationService service provides catalog replication methods. Table 32 shows the supported methods.

Table 32.   ReplicationService Methods

Method Names	Description
export() Method on page 37	Exports catalog changes to a specified log file.
import() Method on page 38	Import changes from the log file.
markForReplication() Method on page 39	Change the "replicable" flag on a specified folder and its descendants.
purgeLog() Method on page 39	Clean replication the specified logs.

export() Method

Exports catalog changes to a specified log file.

Signature

void export (String filename, CatalogItemsFilter filter, bool bExportAll, String sessionID);

Argument	Description
String filename	The name of the log file.
CatalogItemsFilter filter	Defines the subset of changes to be exported. The filter.items field cannot be null.

Argument	Description
bool bExportAll	When TRUE then the contents of folders specified in filter.items and their descendants are written to the export file as if they were inserted right before that method was called. The filter's to and from date fields are ignored.
String sessionID	A string value that contains the session ID to log off from the SOAP session. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

import() Method

Import changes from the log file.

Signature

ImportError[] import (String importFilePath, DateTime lastPurgedLog, bool updateReplicationLog, bool returnErrors, CatalogItemsFilter filter, String sessionID);

Argument	Description
String importFilePath	The path of the log file to import.
DateTime lastPurgedLog	The date and time of when 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.
bool updateReplicationLog	If FALSE then the replication log is not updated.
bool returnErrors	If TRUE then the function returns an array of ImportError objects which describes cases when changes recorded in the import file which satisfy filter conditions were not replayed.
CatalogItemsFilter filter	Used to filter changes made within a particular time period, and to catalog items in specified folders. Can be null.
String sessionID	A string value that contains the session ID to log off from the SOAP session. The session ID is usually returned by the logon call. 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 17.

markForReplication() Method

Changes the "replicable" flag on a specified folder and its descendants.

Signature

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

Argument	Description
String item	The path of the folder.
bool replicate	To mark the folder as replicable, set this to TRUE. To remove the replicable flag, set this to FALSE.
String sessionID	A string value that contains the session ID to log off from the SOAP session. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

purgeLog() Method

Clean replication the specified logs.

Signature

void purgeLog (String[] items, DateTime timestamp, String sessionID);

Argument	Description
String[] items	List of folder paths to clean.
DateTime timestamp	Cleans only those log items where the last modified time is earlier than the timestamp
String sessionID	A string value that contains the session ID to log off from the SOAP session. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

ReportEditingService Service

The ReportEditingService service merges arguments and Siebel Analytics Web data to create and return the results. Table 33 shows the supported methods.

Table 33.   ReportEditingService Methods

Method Names	Description
applyReportParams() Method on page 40	Applies report arguments to the report object and returns the results.
generateReportSQL() Method on page 41	Retrieves the SQL query for a given report.

applyReportParams() Method

The applyReportParams() method applies report arguments to the report and returns the results.

Signature

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

Arguments	Description
ReportRef object	The path to the report definition, supplied in the ReportRef common structure. For information about the ReportRef structure, read "ReportRef Structure" on page 21.
ReportParams reportParams	Optional. The filters or variables to apply to the report before execution, supplied in the ReportParams common structure. For more information, read "Description of Siebel Analytics Web SOAP API Methods" on page 27.
boolean encodeInString	A boolean value, 1 (true) or 0 (false). When set to true, the returned report object is encoded as a character string.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. 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

The generateReportSQL() method retrieves the logical SQL query for a given report.

Signature

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

Arguments	Description
ReportRef reportRef	The path to the report definition supplied in the ReportRef common structure. For more information, read "ReportRef Structure" on page 21.
ReportParams reportParams	Optional. 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 Siebel Analytics Web SOAP API Methods" on page 27.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. 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

The SAWSessionService service provides authentication methods such as logon and logoff, and other session-related methods. Table 34 shows the supported methods.

Table 34.   SAWSessionService Methods

Method Name	Description
getCurUser() Method on page 42	Gets the current user ID for the session.
impersonate() Method on page 42	Logs on and then impersonates the user.
impersonateex() Method on page 43	Logs on and then impersonates the user. Similar to the impersonate method, but impersonateex can specify optional session parameters.
keepAlive() Method on page 44	Instructs Siebel Analytics Web not to end particular sessions due to inactivity.
logoff() Method on page 44	Logs the user off Siebel Analytics Web.

Table 34.   SAWSessionService Methods

Method Name	Description
logon() Method on page 44	Authenticates the user.
logonex() Method on page 45	Authenticates the user. Similar to the logon method, but logonex can specify optional session parameters.

getCurUser() Method

The getCurUser() method gets the current user name for the session.

Signature

String getCurUser(String sessionID);

Argument	Description
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. 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

The impersonate() method in the SAWSessionService service logs on and impersonates the user. 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	A character string that contains the user name to log on and authenticate.

Arguments	Description
String password	A character string that contains the password for the user. If there is no password for the user, leave this field empty (void).
String impersonateID	A character string that contains 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 SOAP API calls to identify your SOAP session.

impersonateex() Method

The impersonateex() method in the SAWSessionService service logs on and impersonates the user. 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, an exception is thrown.

Signature

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

Arguments	Description
String name	A character string that contains the user name to log on and authenticate.
String password	A character string that contains the password for the user. If there is no password for the user, leave this field empty (void).
String impersonateID	A character string that contains the user name to impersonate the authenticated user.
SAWSessionParameters sessionparams	Optional. The sessionparams to use, supplied in the SAWSessionParameters structure. For information about the SAWSessionParameters structure, read "SAWSessionParameters Structure" on page 25.

Returns

This method returns the session ID and sets an HTTP session cookie. The session ID is used in other SOAP API calls to identify your SOAP session.

keepAlive() Method

The keepAlive() method instructs Siebel Analytics Web not to end particular Siebel Analytics Web user sessions due to inactivity. The effect of this call on session lifetime is the same as if those users performed an activity in the browser such as clicking a report, or invoking a SOAP call. For more information about ending Siebel Analytics Web user sessions due to inactivity, read the topic "Setting the Time to Log Users Off Siebel Analytics Web Automatically" in the *Siebel Analytics Web Administration Guide*.

Signature

void keepAlive(String[] sessionIDs);

Argument	Description
String[] sessionIDs	An array of character strings that contains the session IDs to remain logged on.

logoff() Method

The logoff() method logs the user off Siebel Analytics Web.

Signature

void logoff(String sessionID);

Argument	Description
String sessionID	A string value that contains the session ID to log off from the SOAP session. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

logon() Method

The logon() method authenticates the user. If authentication fails, an exception is thrown.

Signature

String logon(String username, String password)

Arguments	Description
String username	A character string that contains the user name to authenticate.
String password	A character string that contains 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 SOAP API calls to identify your SOAP session.

logonex() Method

The logonex() method authenticates the user. Similar to the logon method, but logonex can specify optional session parameters. If authentication fails, an exception is thrown.

Signature

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

Arguments	Description
String username	A character string that contains the user name to authenticate.
String password	A character string that contains the password for the user. If there is no password, leave this field empty (void).
SAWSessionParameters sessionparams	Optional. The sessionparams to use, supplied in the SAWSessionParameters structure. For information about the SAWSessionParameters structure, read "SAWSessionParameters Structure" on page 25.

Returns

This method returns the session ID and sets an HTTP session cookie. The session ID is used in other SOAP API calls to identify your SOAP session.

SecurityService Service

The SecurityService service provides methods for identifying accounts and privileges. Table 35 shows the supported methods.

Table 35. SecurityService Methods

Method Names	Description
forgetAccount() Method on page 46	Removes a Siebel Analytics Web internal ID to account name mapping.
getGlobalPrivilegeACL() Method on page 46	Get the Access Control List for global privileges.
getGlobalSAWPrivileges() Method on page 47	Get the list of all global privileges.
updateGlobalPrivilegeACL() Method on page 47	Update the Access Control List for global privileges.

forgetAccount() Method

Removes a Siebel Analytics Web 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 call with a misspelled account name.

Signature

void forgetAccount(Account account);

Argument	Description
Account account	The accounts to forget, supplied in the Account structure. For information about the Account structure, read "Account Structure" on page 15.

getGlobalPrivilegeACL() Method

Get the Access Control List for global privileges.

Signature

ACL getGlobalPrivilegeACL(String privilegeName, String sessionID);

Argument	Description
String privilegeName	String containing the name of privilege to get.
String sessionID	A string value that contains the session ID to log off from the SOAP session. The session ID is usually returned by the logon call. 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 15.

getGlobalSAWPrivileges() Method

Get the list of all global privileges.

Signature

Privilege[] getGlobalSAWPrivileges(String sessionID);

Argument	Description
String sessionID	A string value that contains the session ID to log off from the SOAP session. The session ID is usually returned by the logon call. 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 a Privilege structure. For more information on the Privilege structure, read "Privilege Structure" on page 19.

updateGlobalPrivilegeACL() Method

Update the Access Control List for global privileges.

Signature

void updateGlobalPrivilegeACL(String SAWPrivilegeName, ACL acl, UpdateACLParams options, String sessionID);

Arguments	Description
String SAWPrivilegeName	String containing the name of privilege to update.
ACL acl	The Access Control List to update, supplied in the ACL structure. For information about the ACL structure, read "ACL Structure" on page 15.
UpdateACLParams options	The Access Control List parameters to update, supplied in the UpdateACLParams structure. For information about the UpdateACLParams structure, read "UpdateACLParams Structure" on page 26.
String sessionID	A string value that contains the session ID to log off from the SOAP session. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

WebCatalogService Service

The WebCatalogService service provides methods for navigating and managing the Web Catalog, and reading and writing Web Catalog objects in XML format. Table 36 shows the supported methods.

Table 36.   WebCatalogService Methods

Method Names	Description
copyItem() Method on page 49	Copies an object from one location to another in the Web Catalog.
createFolder() Method on page 49	Creates a new folder in the Web Catalog.
createLink() Method on page 50	Creates a link to the Web Catalog.
deleteItem() Method on page 50	Deletes an object from the Web Catalog.
getItemInfo() Method on page 51	Gets Web Catalog information for an object.
getSubItems() Method on page 51	Gets the collection of child subitems for an object in the Web Catalog.
moveItem() Method on page 52	Moves an object in the Web Catalog to a different location in the catalog.
readObject() Method on page 53	Reads an object from the Web Catalog.
removeFolder() Method on page 53	Deletes a folder from the Web Catalog.
setItemAttributes() Method on page 54	Sets attribute flags for the specified catalog item.

Table 36.   WebCatalogService Methods

Method Names	Description
setItemProperty() Method on page 54	Sets a property for an object in the Web Catalog.
takeOwnership() Method on page 54	Take ownership on the specified item.
writeObject() Method on page 55	Writes an object to the Web Catalog.
writeReport() Method on page 55	Writes a set of results to the Web Catalog.
writeDashboard() Method on page 56	Writes a dashboard object to the Web Catalog.
writeDashboardPrompt() Method on page 57	Writes a dashboard prompt to the Web Catalog.
writeDashboardPage() Method on page 58	Writes a dashboard page to the Web Catalog.
writeSavedFilter() Method on page 58	Writes a filter to the Web Catalog.

copyItem() Method

The copyItem() method copies an object from one location in the Web Catalog to another location in the Web Catalog.

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

Arguments	Description
String pathSrc	The current path to the object in the Web Catalog.
String pathDest	The location in the Web Catalog where the object should be copied.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

createFolder() Method

The createFolder() method creates a new folder in the Web Catalog.

Signature

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

Arguments	Description
String path	The location in the Web Catalog where the folder should be created, including the name of the new folder.
boolean createIfNotExists	A boolean value, 1 (true) or 0 (false). When set to true, the folder object is created in the Web Catalog if it does not already exist. When set to false, the folder object is not recreated if it already exists.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

createLink() Method

The createLink() method creates a link to the Web Catalog.

Signature

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

Arguments	Description
String sPath	The path to the parent object in the Web Catalog.
String sTargetPath	The location in the Web Catalog to which the link being created should refer.
boolean overwriteIfExists	A boolean value, 1 (true) or 0 (false). When set to true, the link is overwritten if it already exists in the Web Catalog. When set to false, the link is not overwritten if it already exists in the Web Catalog.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

deleteItem() Method

The deleteItem() method deletes an object from the Web Catalog. To delete a folder, read

Signature

void deleteItem(String path, String sessionID);

Arguments	Description
String path	The path to the object in the Web Catalog.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

getItemInfo() Method

The getItemInfo() method gets Web Catalog information for an object.

Signature

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

Arguments	Description
String path	The path to the object in the Web Catalog.
boolean resolveLinks	A boolean value, 1 (true) or 0 (false). When set to true, and the path in the Web Catalog refers to a link, Analytics retrieves information for the object pointed to by the link.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns Web Catalog information for an object in an ItemInfo structure. For more information, read "ItemInfo Structure" on page 18.

getSubItems() Method

The getSubItems() method gets the collection of child subitems for an object in the Web Catalog.

Signature

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

Arguments	Description
String path	The path to the parent object in the Web Catalog.
String mask	The 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	A boolean value, 1 (true) or 0 (false). When set to true, and the path in the Web Catalog refers to a link, information is retrieved for the child subitems of the object pointed to by the link.
GetSubItemsParams options	Optional parameters supplied in the GetSubItemsParams structure. For information about the GetSubItemsParams structure, read "GetSubItemsParams Structure" on page 17.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. 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 18.

moveItem() Method

The moveItem() method moves an object in the Web Catalog to a different location in the Web Catalog.

Signature

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

Arguments	Description
String pathSrc	The current path to the object in the Web Catalog.
String pathDest	The location in the Web Catalog where the object should be moved.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

readObject() Method

The readObject() method reads an object from the Web Catalog and returns CatalogObject structure.

Signature

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

Arguments	Description
String path	The location where the retrieved object should be placed.
boolean returnXmlString	A boolean value, 1 (true) or 0 (false). When set to true, the catalogObject field returned in the CatalogObject structure is a character string that contains the XML representation of the object stored in the Web Catalog. When set to false, the SOAP client needs to analyze the xsi:type attribute of the root node of the returned XML to determine the type of object to create. If the xsi:type attribute is unknown, an exception may be thrown depending on the SOAP client's implementation.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. 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 Web Catalog. For a description of the CatalogObject structure, read "CatalogObject Structure" on page 16.

removeFolder() Method

The removeFolder() method deletes a folder and its contents from the Web Catalog. To delete an object other than a folder and its contents, read "deleteItem() Method" on page 50.

Signature

void removeFolder(String path, String sessionID);

Arguments	Description
String path	The path to the folder in the Web Catalog.
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

setItemAttributes() Method

The setItemAttributes() method sets attribute flags for a specified catalog item.

Signature

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

Arguments	Description
String path	The path to the folder in the Web Catalog.
int attributes	Combination of the following flags: 1 = read only 2 = archive 4 = hidden 8 = system
String sessionID	A character string that identifies the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

setItemProperty() Method

The setItemProperty() method sets a property for an object in the Web Catalog.

Signature

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

Arguments	Description
String path	The path to the object in the Web Catalog.
String name	A character string that contains the name of the property to set.
String value	A character string that contains the new setting for the property.
String sessionID	A character string that contains the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

takeOwnership() Method

Take ownership on the specified item.

Signature

void takeOwnership(String path, String sessionID);

Arguments	Description
String path	The location in the Web Catalog of the object to take ownership.
String sessionID	A string value that contains the session ID to log off from the SOAP session. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeObject() Method

The writeObject() method writes an object to the Web Catalog in XML format.

Signature

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

Arguments	Description
CatalogObject object	The object to write to the Web Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 16. 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	The location in the Web Catalog where the object should be written.
boolean resolveLinks	A boolean value, 1 (true) or 0 (false). When set to true, and the path in the Web Catalog refers to a link, the object is written to the location pointed to by the link.
boolean allowOverwrite	A boolean value, 1 (true) or 0 (false). When set to true, if the object already exists in the Web Catalog, it is overwritten. When set to false, if the object already exists in the Web Catalog, it is not overwritten.
String sessionID	A character string that contains the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeReport() Method

The writeReport() method writes a set of results to the Web Catalog.

Signature

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

Arguments	Description
CatalogObject object	The object to write to the Web Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 16. 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	The location in the Web Catalog where the results should be written.
boolean resolveLinks	A boolean value, 1 (true) or 0 (false). When set to true, and the path in the Web Catalog refers to a link, the results are written to the location pointed to by the link.
boolean allowOverwrite	A boolean value, 1 (true) or 0 (false). When set to true, if the results already exist in the Web Catalog, they are overwritten. When set to false, if the results already exist in the Web Catalog, they are not overwritten.
String sessionID	A character string that contains the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeDashboard() Method

The writeDashboard() method writes a dashboard object to the Web Catalog.

Signature

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

Arguments	Description
CatalogObject object	The dashboard object to write to the Web Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 16. 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	The location in the Web Catalog where the dashboard object should be written.

Arguments	Description
boolean resolveLinks	A boolean value, 1 (true) or 0 (false). When set to true, and the path in the Web Catalog refers to a link, the dashboard object is written to the location pointed to by the link.
boolean allowOverwrite	A boolean value, 1 (true) or 0 (false). When set to true, if the dashboard object already exists in the Web Catalog, it is overwritten. When set to false, if the dashboard object already exists in the Web Catalog, it is not overwritten.
String sessionID	A character string that contains the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeDashboardPrompt() Method

The writeDashboardPrompt() method writes a dashboard prompt to the Web Catalog.

Signature

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

Arguments	Description
CatalogObject object	The dashboard prompt object to write to the Web Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 16. 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	The location in the Web Catalog where the dashboard prompt should be written.
boolean resolveLinks	A boolean value, 1 (true) or 0 (false). When set to true, and the path in the Web Catalog refers to a link, the dashboard prompt is written to the location pointed to by the link.
boolean allowOverwrite	A boolean value, 1 (true) or 0 (false). When set to true, if the dashboard prompt already exists in the Web Catalog, it is overwritten. When set to false, if the dashboard prompt already exists in the Web Catalog, it is not overwritten.
String sessionID	A character string that contains the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeDashboardPage() Method

The writeDashboardPage() method writes a dashboard page to the Web Catalog.

Signature

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

Arguments	Description
CatalogObject object	The dashboard page object to write to the Web Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 16.
	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	The location in the Web Catalog where the dashboard page should be written.
boolean resolveLinks	A boolean value, 1 (true) or 0 (false). When set to true, and the path in the Web Catalog refers to a link, the dashboard page is written to the location pointed to by the link.
boolean allowOverwrite	A boolean value, 1 (true) or 0 (false). When set to true, if the dashboard page already exists in the Web Catalog, it will be overwritten. When set to false, if the dashboard page already exists in the Web Catalog, it will not be overwritten.
String sessionID	A character string that contains the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

writeSavedFilter() Method

The writeSavedFilter() method i writes a filter to the Web Catalog.

Signature

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

Arguments	Description
CatalogObject object	The filter object to write to the Web Catalog, supplied in the CatalogObject structure. For information about the CatalogObject structure, read "CatalogObject Structure" on page 16. 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	The location in the Web Catalog where the filter should be written.
boolean resolveLinks	A boolean value, 1 (true) or 0 (false). When set to true, and the path in the Web Catalog refers to a link, the filter is written to the location pointed to by the link.
boolean allowOverwrite	A boolean value, 1 (true) or 0 (false). When set to true, if the filter already exists in the Web Catalog, it is overwritten. When set to false, if the filter already exists in the Web Catalog, it is not overwritten.
String sessionID	A character string that contains the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

XMLViewService Service

The XMLViewService service retrieves results from Siebel Analytics Web in XML format. Table 37 shows the supported method.

Table 37.   XMLView Service Method

Method Name	Description
"getResults() Method"	Returns the report results, in XML format.

getResults() Method

The getResults() method returns the report results, in XML format.

Signature

Object getResults(ReportRef report, String outputFormat, boolean encodeInString, ReportParams reportParams, String sessionID);

Arguments	Description
ReportRef report	The report definition, supplied in the ReportRef common structure. For more information, read "ReportRef Structure" on page 21.
String outputFormat	A character string that identifies the output format of the returned report object. Valid values are: ■ "urn: SAWRowset.data" This value returns rows without metadata. ■ "urn: SAWRowset.schema" This value returns metadata only. ■ "urn: SAWRowset.both" This value returns both rows and metadata.
boolean encodeInString	A boolean value, 1 (true) or 0 (false). When set to true (the usual value), it indicates that the returned report object is encoded as a character string.
ReportParams reportParams	Optional. 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 20.
String sessionID	A character string that contains the session ID. The session ID is usually returned by the logon call. If the SOAP client engine can handle HTTP cookies, you can omit the session ID or set it to null.

Returns

Returns the XML results for the specified report definition.

5 Format of Returned Recordsets

This is the basic structure for Siebel Analytics Web rowset XML output:

```
<Recordset xmls="Siebel Analytics 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 the Siebel Analytics Web SOAP API to extract Web 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.

```csharp
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