Siebel Analytics Web Administration Guide

Version 7.8.2

May 2005

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404
Copyright © 2005 Siebel Systems, Inc.
All rights reserved.
Printed in the United States of America

No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior agreement and written permission of Siebel Systems, Inc.

Siebel, the Siebel logo, UAN, Universal Application Network, Siebel CRM OnDemand, TrickleSync, Universal Agent, and other Siebel names referenced herein are trademarks of Siebel Systems, Inc., and may be registered in certain jurisdictions.

Other product names, designations, logos, and symbols may be trademarks or registered trademarks of their respective owners.

PRODUCT MODULES AND OPTIONS. This guide contains descriptions of modules that are optional and for which you may not have purchased a license. Siebel's Sample Database also includes data related to these optional modules. As a result, your software implementation may differ from descriptions in this guide. To find out more about the modules your organization has purchased, see your corporate purchasing agent or your Siebel sales representative.

U.S. GOVERNMENT RESTRICTED RIGHTS. Programs, Ancillary Programs and Documentation, delivered subject to the Department of Defense Federal Acquisition Regulation Supplement, are "commercial computer software" as set forth in DFARS 227.7202, Commercial Computer Software and Commercial Computer Software Documentation, and as such, any use, duplication and disclosure of the Programs, Ancillary Programs and Documentation shall be subject to the restrictions contained in the applicable Siebel license agreement. All other use, duplication and disclosure of the Programs, Ancillary Programs and Documentation by the U.S. Government shall be subject to the applicable Siebel license agreement and the restrictions contained in subsection (c) of FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987), or FAR 52.227-14, Rights in Data—General, including Alternate III (June 1987), as applicable. Contractor/licensor is Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404.

Proprietary Information

Siebel Systems, Inc. considers information included in this documentation and in Siebel Business Applications Online Help to be Confidential Information. Your access to and use of this Confidential Information are subject to the terms and conditions of: (1) the applicable Siebel Systems software license agreement, which has been executed and with which you agree to comply; and (2) the proprietary and restricted rights notices included in this documentation.

Contents

Chapter 6: Administering the Siebel Analytics Web Catalog

Chapter 7: Managing Siebel Analytics Web Security

Chapter 8: Using Siebel Analytics Web Logging

Chapter 9: Working with Data and Managing the Web Catalog Using Siebel Analytics Web SOAP API

1 What's New in This Release

What's New in Siebel Analytics Web Administration Guide, Version 7.8.2

Table 1 lists changes described in this version of the documentation to support Release 7.8.2 of the software.

Table 1. New Product Features in Siebel Analytics Web Administration Guide, Version 7.8.2

Topic	Description
"Blocking Requests in Siebel Answers" on page 39.	New topic. Administrators can block specific requests, such as requiring users to include certain columns with others, or requiring filters when certain columns are requested.
"Specifying View Defaults for Siebel Answers and Siebel Intelligence Dashboard Users" on page 43.	New topic. Administrators can control some aspects of the initial state of new views that are added to a request within Siebel Answers and new objects that are added to a dashboard page.
"Viewing Information About Active Siebel Delivers iBot Sessions" on page 53.	New topic. Administrators can view information about currently active iBot sessions triggered by the Scheduler, such as a list of active iBots per session and the recipients for each active iBot.
"Controlling Access to Saved Selection Options in Siebel Intelligence Dashboards" on page 60.	New topic. It provides an overview of saved selections and information about administering saved selections.

The features listed in Table 2 were introduced in Version 7.8.1, the Limited Distribution release of the software.

Table 2. New Product Features in Siebel Analytics Web Administration Guide, Version 7.8.1

Topic	Description
Creating an action link. "Creating Siebel Intelligence Dashboards Action Links" on page 58.	Corrected the HTML in Step 5 of the procedure to create an action link.
Javahost service. "Using the Javahost Service in Siebel Analytics Web" on page 25.	Added information on configuring the Javahost service.

Table 2. New Product Features in Siebel Analytics Web Administration Guide, Version 7.8.1

Topic	Description
Making configuration changes. "Making Siebel Analytics Web Configuration Changes" on page 11.	Removed all references to working in the Windows registry. Beginning with this version of Siebel Analytics, you should make Siebel Analytics Web configuration changes in the configuration file instanceconfig.xml and not in the Windows registry.
SAW Replication. "Replicating Siebel Analytics Web Catalogs" on page 80.	Added information on how to replicate a Web Catalog from one Siebel Analytics Web Server to other servers.
Server Logging. "Using Siebel Analytics Web Logging" on page 123.	Added information on Siebel Analytics Web Server logging.
Siebel Analytics Web ReportUI Portlet. "Configuring the Siebel Analytics Web ReportUI Portlet" on page 204.	Added information on how to use Siebel Analytics Web reports in a WebSphere Portlet.
SOAP API. "Working with Data and Managing the Web Catalog Using Siebel Analytics Web SOAP API" on page 141.	Added information on using the SOAP API.
Web Catalog Backup. "How the Siebel Analytics Web Catalog Backup Process Works" on page 70.	Revised information on the improved Web Catalog backup parameters.

sw

2 Administering Siebel Analytics Web

This chapter provides general post-installation configuration and administration procedures that are *not* specific to Siebel Answers, Siebel Delivers, Siebel Intelligence Dashboards, or Siebel Analytics Web Catalog. Directions for configuring these components of Siebel Analytics Web are given in subsequent chapters.

This chapter contains the following topics:

Making Siebel Analytics Web Configuration Changes

This section provides procedures for making configuration changes. You need to make changes only if you want to change default parameters, such as the name of the Siebel Analytics Web Catalog, or override internal default settings, such as the time for client connections to expire.

You make configuration changes by modifying the Siebel Analytics Web XML message file instanceconfig.xml, which holds configuration settings for Siebel Analytics Web.

NOTE: If you have previously made configuration changes by modifying the Windows registry, you should migrate those changes to the XML configuration file instanceconfig.xml. In the Windows registry, entries under the Common key remain valid.

For general background information about customizing the Siebel Analytics Web XML files, see "Customizing the Siebel Analytics Web User Interface Using XML Message Files" on page 196.

To make changes to the configuration file instanceconfig.xml

1. Navigate to Siebel Analytics data directory at the following location:

   `\SiebelAnalyticsData\Web\Config`

   CAUTION: Always make a backup copy of the file instanceconfig.xml before you make any changes.

2. Locate the file instanceconfig.xml and make a backup copy.
3. Use a text editor to open the file instanceconfig.xml.
4. Place your entries between the tags <ServerInstance> and </ServerInstance>, using the values given elsewhere in this chapter.
5. Save the file when you are done.

   Your changes take effect when the Analytics Web Server service is restarted.

Example instanceconfig.xml File

The following XML file is an example of the instanceconfig.xml file:

```
<?xml version="1.0" ?>
<!-- Siebel Analytics Web Configuration File -->
<!-- The following example shows initialization settings for a server instance. -->

<WebConfig>
   <ServerInstance>
      <CatalogPath>/SiebelAnalyticsData/web/catalog/default.webcat<CatalogPath>
      <DSN>AnalyticsWeb</DSN>
   </ServerInstance>
</WebConfig>
```

Several entries are present in the file instanceconfig.xml by default, including the path to the Siebel Analytics Web Catalog, and the name of the Siebel Analytics Server data source name used by Siebel Analytics Web to access Siebel Analytics Server.

For example, the path to the Siebel Analytics Web Catalog is shown between the <CatalogPath> and </CatalogPath> tags:

- In Windows, an example path might be the following:

  `<CatalogPath>c:\SiebelAnalyticsData\Web\Catalog\default.webcat</CatalogPath>`

- In UNIX, an example path might be the following:

`<CatalogPath>/usr/local/SiebelAnalyticsData/web/catalog/default.webcat</CatalogPath>`

Changing the Siebel Analytics Web ODBC DSN

Siebel Analytics Web accesses a Siebel Analytics Server repository with a single Siebel Analytics Server data source name (DSN). The installation process configures a DSN named Analytics Web for this purpose. For information on configuring a new or existing DSN, see *Siebel Analytics Server Administration Guide*.

Siebel Answers must use a single DSN.

If you change the name of the DSN, you need to update the Siebel Analytics Web configuration file instanceconfig.xml to use the new name.

The following entry is an example:

```
<DSN>Analytics Web Production</DSN>
```

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Setting the Siebel Analytics Web Configuration File Path

You can override the path that Siebel Analytics Web searches for configuration files upon startup. The internal default location is $(DataDir)\Web\Config, where $(DataDir) is the resolved entry of one of the following specifications:

- In Windows:

  ```
  HKEY_LOCAL_MACHINE\SOFTWARE\Siebel Systems, Inc.\Siebel
  Analytics\Common\7.7\DataDir
  ```

- In UNIX, this is configured by an environment variable defined in initialization scripts. For more information, see *Siebel Analytics Installation and Configuration Guide*.

Siebel Analytics Web must have read permission to this path. The default data directory is SiebelAnalyticsData.

You can override the internal default by adding the following entry to the Siebel Analytics Web configuration file instanceconfig.xml:

```
<ConfigDir>SiebelAnalyticsTest</ConfigDir>
```

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Disabling the Siebel Analytics Web Feature to Remember Users' Names and Passwords

By default, users are asked if they want to have their user names and passwords remembered when logging on to Siebel Analytics Web. You can disable this behavior by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. When the value is set to No, users always have to type their user names and passwords.

The following entry is an example:

```
<AllowRememberPassword>No</AllowRememberPassword>
```

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Setting the Time to Expire for Idle Siebel Analytics Web Client Connections

The entry ConnectionExpireMinutes defines the length of idle time that the connection between the Siebel Analytics Web Server and the Siebel Analytics Server is maintained before the connection is closed. You can override the number of minutes to elapse by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. The internal default is 3 minutes.

The following entry is an example:

```
<ConnectionExpireMinutes>3</ConnectionExpireMinutes>
```

When a user logs into the Analytics application, a connection is created from the browser client to the Siebel Analytics Web Server, and another connection is created from the Siebel Analytics Web Server to the Siebel Analytics Server. If the user's session remains idle for three minutes or more (the user does not perform any operation), the connection from the Siebel Analytics Web Server to the Siebel Analytics Server is closed or dropped. The connection from the browser to the Siebel Analytics Web Server still remains intact and has no noticeable impact to the user. The next time the user performs an operation such as running a report or navigating to a dashboard, a new connection is created from the Siebel Analytics Web Server to the Siebel Analytics Server.

NOTE: This setting affects idle time only. For example, if the user were to run a request that executes for more than three minutes, the original connection from the Siebel Analytics Web Server to the Siebel Analytics Server remains intact throughout, until the three-minute idle time is encountered again. This setting therefore keeps down the number of open connections to the server.

You can also set the amount of time to elapse before a user is logged off automatically. For more information, see "Setting the Time to Log Users Off Siebel Analytics Web Automatically" on page 18.

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Administering Sessions in Siebel Analytics Web

To perform session administration, you must have the Manage Sessions privilege. For more information, see "Default Siebel Analytics Web Privilege Assignments" on page 114.

To view information about logged on users and running requests

1. In Siebel Analytics Web, do one of the following:
   - If you are running Siebel Analytics stand-alone, click the Admin link.
   - If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.
2. Click the Manage Sessions link.

   The Sessions table of the Sessions area gives information about users who are logged on, including the unique Session ID which is based on the IP address of the connected computer.

   The Cursor Cache table shows the status of requests made by users.

Field	Description
User	The name of the user who ran the request and last placed it into the cache. If two IDs are shown, the second ID is impersonating the first. For example, the entry lhurley/administrator means that the Administrator account logged on and impersonated lhurley. This can happen when the Scheduler starts an iBot on someone's behalf. The Scheduler logs on and impersonates that user so security and content filters still apply.
Refs	The number of references to this entry since it was placed into the cache.
Status	The status of the request using this cache entry: ■ **Running.** The request is currently running. ■ **Finished.** The request has finished. ■ **Queued.** The system is waiting for a thread to become available so the request can be processed. ■ **Canceling.** The system is in the process of canceling the request. ■ **Error.** An error was encountered during the processing or running of the request. Look in the Statement column for information about the error.
Time	The time taken to process and run the request, displayed in one second increments. A value of 0s (zero seconds) indicates that the request took under 1 second to complete.
Statement	The SQL issued for the request, or if the request resulted in an error, information about the nature of the error.

To cancel all running requests

1. In Siebel Analytics Web, do one of the following:
   - If you are running Siebel Analytics stand-alone, click the Admin link.
   - If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.
2. Click the Manage Sessions link.
3. Click the Cancel Running Requests link.
4. Click Finished.

To cancel one running request

1. In Siebel Analytics Web, do one of the following:
   - If you are running Siebel Analytics stand-alone, click the Admin link.
   - If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.
2. Click the Manage Sessions link.
3. In the Cursor Cache table, identify the request and click the Cancel link in the Action column.

   The user receives a message indicating that the request was canceled by a Siebel administrator.

To clear the Web cache

1. In Siebel Analytics Web, do one of the following:
   - If you are running Siebel Analytics stand-alone, click the Admin link.
   - If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.
2. Click the Manage Sessions link.
3. In the Cursor Cache table, identify the request and click the link Close All Cursors.
4. Click Finished.

To clear the cache entry associated with a request

1. In Siebel Analytics Web, do one of the following:
   - If you are running Siebel Analytics stand-alone, click the Admin link.
   - If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.
2. Click the Manage Sessions link.
3. In the Cursor Cache table, identify the request and click the Close link in the Action column.

To view the query file for information about a request

1. In Siebel Analytics Web, do one of the following:
   - If you are running Siebel Analytics stand-alone, click the Admin link.
   - If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.
2. Click the Manage Sessions link.
3. In the Cursor Cache table, identify the request and click the View Log link.

NOTE: Query logging must be turned on for data to be saved in this log file. For more information about query logging, see *Siebel Analytics Server Administration Guide*.

Setting the Time to Expire for Siebel Analytics Web Client Sessions

The entry ClientSessionExpireMinutes defines the length of idle time that can elapse before the Siebel Analytics Web Server removes the user's client (browser) session information from its memory. This session includes user-specific state information such as request cache, dashboard page state, subject area information, connection information, and so on.

The internal default is 1440 (24 hours).

For example, if the user does not access Siebel Analytics Web in 24 hours or more, the server's information about that session is removed completely, in which case the user will be logged out of the application and will need to log back in. All state information is lost.

You can override the number of minutes to elapse before browser client sessions are removed by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. The value must be greater than or equal to the value specified for the setting SearchIDExpireMinutes, described in "Setting the Time to Keep an Unsaved Siebel Analytics Request" on page 17.

The following entry is an example:

```
<ClientSessionExpireMinutes>1440</ClientSessionExpireMinutes>
```

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Setting the Time to Keep an Unsaved Siebel Analytics Request

You can override the time, in minutes, that an unsaved request remains valid by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. This entry applies to unsaved requests only. The internal default is 180 (three hours).

NOTE: The value must be less than or equal to the value specified for the setting ClientSessionExpireMinutes, described in "Setting the Time to Expire for Siebel Analytics Web Client Sessions" on page 17.

The following entry is an example:

```
<SearchIDExpireMinutes>1440</SearchIDExpireMinutes>
```

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Setting the Time to Cancel an Unattended Siebel Analytics Request

You can override the time to elapse, in minutes, before an unattended request is cancelled by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. An unattended request is one that has not been accessed in the number of minutes specified by this setting. The internal default is 5. The minimum value is 2.

This entry handles the case where a user is at the Request screen in Siebel Answers and browses elsewhere, abandoning the request, at least temporarily. Do not set the value too small, however, as the user may return to the request.

The following entry is an example:

```
<UnaccessedRunningTimeoutMinutes>5</UnaccessedRunningTimeoutMinutes>
```

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Setting the Time to Log Users Off Siebel Analytics Web Automatically

You can override the time to elapse, in minutes, before a user is automatically logged off. This setting applies only to users who do not have the following option selected at the Log In screen:

Remember my ID and password

If this value expires before the value set in the entry ConnectionExpireMinutes, the user can log back in to the existing session. For information about the ConnectionExpireMinutes entry, see "Setting the Time to Expire for Idle Siebel Analytics Web Client Connections" on page 14.

The default is 180 (three hours). You can change the number of minutes by modifying the Siebel Analytics Web configuration file instanceconfig.xml.

The following entry is an example:

```
<LogonExpireMinutes>180</LogonExpireMinutes>
```

NOTE: You can disable this setting by setting it to a value greater than the value for the setting ClientSessionExpireMinutes, described in "Setting the Time to Expire for Siebel Analytics Web Client Sessions" on page 17.

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Managing the Siebel Analytics Web Cache Settings

The entries described in this section are available for managing the Siebel Analytics Web cache:

The cache is accessed when users make requests in Siebel Answers. This is not the same cache that is accessed by the Siebel Analytics Server. You can change the internal defaults by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entries.

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Specifying the Maximum Amount of Time an Entry Can Exist in the Siebel Analytics Web Cache

You can override the maximum amount of time, in minutes, that an entry in the cache can exist before it is removed. The internal default is 60 (one hour).

Depending on (the number of requests being run, an entry may be removed before the time limit expires.

NOTE: The setting for CacheMinUserExpireMinutes, described in "Specifying the Maximum Amount of Time an Entry Can Exist in the Siebel Analytics Web Cache," can force an entry for a particular user to exist for a longer time than that specified by CacheMaxExpireMinutes.

The following entry is an example:

```
<CacheMaxExpireMinutes>60</CacheMaxExpireMinutes>
```

Specifying the Least Amount of Time an Entry Can Exist in the Siebel Analytics Web Cache

You can override the minimum amount of time, in minutes, that an entry in the cache can exist before it is removed. The internal default is 10.

The following entry is an example:

```
<CacheMinExpireMinutes>10</CacheMinExpireMinutes>
```

Specifying the Least Amount of Time an Entry Can Exist in the Siebel Analytics Web Cache After Use

You can override the minimum amount of time, in minutes, that an entry in the cache can exist after it has been viewed by a user. The internal default is 10.

For example, suppose CacheMaxExpireMinutes is set to 60 minutes. If a user views the entry during the 59th minute, the entry exists for that user for an additional 10 minutes. This allows the user to continue paging through the data without requiring a new request to be run.

The following entry is an example:

```
<CacheMinUserExpireMinutes>10</CacheMinUserExpireMinutes>
```

Specifying the Maximum Number of Siebel Analytics Web Open Record Sets

You can override the maximum number of open record sets that Siebel Analytics Web keeps open at any one time. The internal default is 10. The minimum value is 3. For systems under significant loads, you can increase this value to 500 or 1000.

The following entry is an example:

```
<CacheMaxEntries>100</CacheMaxEntries>
```

Configuring the Siebel Analytics Web Cookie Domain

You can configure the cookie domain by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entries:

- “Specifying Siebel Analytics Web Cookie Domain Information” on page 20
- “Overriding the Siebel Analytics Web Cookie Domain Path” on page 20
- “Specifying the Expiration Date for Persisted Siebel Analytics Web Cookies” on page 21

For information about working in the configuration file instanceconfig.xml, see “Making Siebel Analytics Web Configuration Changes” on page 11.

Specifying Siebel Analytics Web Cookie Domain Information

You can specify domain information for a cookie sent to the browser. There is no default value.

The following entry is an example:

```
<CookieDomain>value</CookieDomain>
```

Overriding the Siebel Analytics Web Cookie Domain Path

You can override the domain path that cookies apply to. The internal default is “/”.

The following entry is an example:

```
<CookiePath>/usr/local/test/cookies</CookiePath>
```

Specifying the Expiration Date for Persisted Siebel Analytics Web Cookies

You can override the date on which persisted cookies expire by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. For example, remembered passwords expire on this date (see "Disabling the Siebel Analytics Web Feature to Remember Users' Names and Passwords" on page 14). The default value is "Tue, 31 Dec 2030 23:59:59 GMT".

The format for the date is "day, dd mon year hh:mm:ss GMT", where:

day	The standard three-letter abbreviation for the day of the week.
dd	The 2-digit day of the month.
mon	The standard three-character abbreviation for the month.
year	The 4-digit year.
hh:mm:ss	The hour, minutes and seconds.
GMT	The time zone, Greenwich Mean Time.

You should not change this entry.

The following entry is an example:

```
<CookieExpire>Tue, 31 Dec 2040 23:59:59 GMT<\CookieExpire>
```

Managing Siebel Analytics Web URL Generation and Resource File Location

You can override how Siebel Analytics Web generates URLs and where resource files are located by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entries:

- "Specifying How Siebel Analytics Web Command URLs Are Generated" on page 22
- "Specifying How Siebel Analytics Web Static URLs Are Generated" on page 22
- "Specifying the Location of Siebel Analytics Web Primary Resource Files" on page 22
- "Specifying the Path to Siebel Analytics Web Primary Resource Files" on page 23
- "Specifying the Location of Siebel Analytics Web Nonprimary Resource Files" on page 23
- "Specifying the Path to Siebel Analytics Web Nonprimary Resource Files" on page 24
- "Specifying Whether Siebel Analytics Web Generates Fully Qualified URLs" on page 24

You need to create the tags <URL> and </URL> after the <ServerInstance> tag, and place your entries between the <URL> and </URL> tags.

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

NOTE: Most of the URL entries are interrelated.

Specifying How Siebel Analytics Web Command URLs Are Generated

You can specify how Siebel Analytics Web generates command URLs.

If you explicitly specify an entry, it must be of the following form:

`protocol://server/`*`virtualpath`*

where virtualpath is the complete virtual path to Siebel Analytics Web. The default is determined separately for each client, based on the URL sent by the client to the Siebel Analytics Web server.

The following entry is an example:

```
<URL>
   <AbsoluteCommandURLPrefix>value</AbsoluteCommandURLPrefix>
</URL>
```

Specifying How Siebel Analytics Web Static URLs Are Generated

You can specify how Siebel Analytics Web generates URLs for static resources such as images, script files, style sheets, and other user-specified files. The default is protocol://server from the URL\AbsoluteCommandURLPrefix setting, described in "Specifying How Siebel Analytics Web Command URLs Are Generated" on page 22.

If you explicitly specify an entry, it must be of this form:

`protocol://server`

If you specify a virtual path, it is removed.

This entry also allows you to designate a separate Web server for delivering static resources, thereby reducing the load on the main Web server. This prefix will be used for the resources that have a fully qualified virtual path of the form '/Path/file'. If a resource file has a relative virtual path of the form 'Path/file', the prefix used is the same used for commands to the Siebel Analytics Web server extension.

The following entry is an example:

```
<URL>
   <ResourceServerPrefix>value</ResourceServerPrefix>
</URL>
```

Specifying the Location of Siebel Analytics Web Primary Resource Files

You can override the physical location of Siebel Analytics Web's primary resource files (these are the resource files distributed with Siebel Analytics Web, not user-customized files such as custom styles or skins). The internal default is $(INSTALL_DIR)\Web\App\Res.

You must provide a full path. Siebel Analytics Web must have read permission to this path. For example, if this is a shared network resource, the Siebel Analytics Web administrator needs to make sure that the user under which Siebel Analytics Web is running has read access to the share, as well as read access to the file system the share is exported from.

NOTE: If the value for this entry is different from the physical location of the Siebel Analytics Web DLLs, you must specify a setting for URL\ResourceVirtualPath, described in "Specifying the Path to Siebel Analytics Web Primary Resource Files" on page 23.

The following entry is an example:

```
<URL>
   <ResourcePhysicalPath>value</ResourcePhysicalPath>
</URL>
```

Specifying the Path to Siebel Analytics Web Primary Resource Files

You can override the virtual path used for Siebel Analytics Web's primary resource files as specified by the setting URL\ResourcePhysicalPath, described in "Specifying the Location of Siebel Analytics Web Nonprimary Resource Files" on page 23. These resource files and customer-defined resource files must be served from the same Web server.

For generating relative URLs, the virtual path defaults to Res, assuming that the resource folder is present under the same virtual directory as the Siebel Analytics Web DLLs.

For generating absolute URLs, the entry of the value URL\AbsoluteCommandURLPrefix is used as the default.

The value must be a fully qualified virtual path of this form:

```
'/VirtualPath'
```

If you omit the leading slash, one will be added.

The following entry is an example:

```
<URL>
   <ResourceVirtualPath>value</ResourceVirtualPath>
</URL>
```

Specifying the Location of Siebel Analytics Web Nonprimary Resource Files

You can override the physical location of resource files that are not part of a default installation. Such resource files include customized styles and skins. The internal default is $(DataDir)\Web\App\Res, where $(DataDir) is the resolved entry of HKEY_LOCAL_MACHINE\SOFTWARE\Siebel Systems, Inc.\Siebel Analytics\Common\7.7\DataDir.

You must provide a full path. Siebel Analytics Web must have read permission to this path. For example, if this is a shared network resource, you need to make sure that the user under which Siebel Analytics Web is running has read access to the share, as well as read access to the file system the share is exported from.

The following entry is an example:

```
<URL>
   <CustomerResourcePhysicalPath>value</CustomerResourcePhysicalPath>
</URL>
```

Specifying the Path to Siebel Analytics Web Nonprimary Resource Files

You can override the virtual path used for resource files that are not part of a default installation as specified in the setting URL\CustomerResourcePhysicalPath, described in "Specifying the Location of Siebel Analytics Web Nonprimary Resource Files" on page 23. The internal default is Res.

The following entry is an example:

```
<URL>
   <CustomerResourceVirtualPath>value</CustomerResourceVirtualPath>
</URL>
```

Specifying Whether Siebel Analytics Web Generates Fully Qualified URLs

You can override whether Siebel Analytics Web always generates fully qualified URLs for resource files that have fully qualified virtual paths. The internal default is No.

When set to No, resources and the Siebel Analytics Web server extension are served from one server. When set to Yes, default resources are served from the same server as the Siebel Analytics Web server extension, and customer resources are served from another server. Depending on the value of the other settings described in this section, you could also set it up to have default and customer resources served from one server, and the Siebel Analytics Web server extension served from another server.

The following entry is an example:

```
<URL>
   <ForceAbsoluteResourceURL>value</ForceAbsoluteResourceURL>
</URL>
```

Specifying the Default Language for the Siebel Analytics Web Login Screen

The default language in which the Siebel Analytics Web login screen appears is obtained from the user's client browser settings. The following procedure explains how to change the language.

NOTE: The following procedure uses Internet Explorer 6.0 as an example. If you are using a different browser, make the necessary substitutions.

To change the default language on a user's login screen

1. In Internet Explorer, choose Tools > Internet Options.

   The Internet Options dialog box appears.

2. Click the Languages button.

   The Language Preference dialog box appears.

   Installed languages appear in the Languages list. The language at the top of the list is used as the default language.

3. If the desired language is not installed on the browser, add it.
4. Use the Move Up and Move Down buttons to position the desired language at the top of the list.
5. Restart the browser and log into Siebel Analytics Web.

   The default language should match the language in the browser's Language list.

   NOTE: If a user does not select a different language from the drop-down list on the login screen, the setting for the User Interface Language at the user's My Account screen determines the language in which the user interface is displayed.

Using the Javahost Service in Siebel Analytics Web

The Javahost service gives Siebel Analytics Web the ability to use functionality provided by Java libraries. It supports the following components:

- Chart generation (Corda)
- PDF generation (FOP)
- SVG renderer (Apache Batik)

Starting and Stopping the Javahost Service

The Javahost service starts and stops automatically, so typically there is no need to start and stop it independently. However, if you need to make configuration changes, you should edit the instanceconfig.xml file, shut down the Javahost service, and then restart it.

The Javahost service has two execution modes:

- **Service mode.** Javahost is running in the background, its console is not visible and the user is not required to be logged on to the computer. This mode is the normally production environment mode.
- **Console mode.** Javahost is running in the user console. This mode is useful for troubleshooting purposes, because it displays messages.

To start or stop the Javahost service in UNIX

- Use the (SAROOT)/web/sawjavahost/bin/saw.sh command:
  - To start it in Service mode, use a "-service" command line switch.
  - To stop the Javahost service, use the (SAROOT)/web/sawjavahost/bin/shutdown.sh command.

  This utility opens a TCP/IP connection to Javahost and sends a shutdown signal to it.

To start or stop the Javahost service in Windows

- To start or stop the Javahost service in Service mode, use the Services control panel and start or stop the Siebel Analytics Javahost Service.
- To start and stop the Javahost service in Console mode requires using the command line:
  - To start it, run <SAROOT>/web/bin/sawjavahostsvc.exe.
  - To stop it, press Ctrl+C.

Javahost Service Command Line Options

When using the command line in UNIX and Windows, you can specify some command line options.

In UNIX, the command line for starting the Javahost service is:

```
run.sh [-h] [-service] [-SawConfigRoot configrootdir]
[-DefaultCordaRoot configrootdir] [-Config instanceconfig]
```

In UNIX, the command line for stopping the Javahost service is:

```
shutdown.sh [-h] [-Config instanceconfig] [-SawConfigRoot configrootdir]
[-Port port] [-Host host]
```

In Windows, the command line for starting the Javahost service in Console mode is:

```
sawjavahostsvc.exe [-regserver | -regserverauto | -unregserver | -h | -service -V ]
[-SawConfigRoot configrootdir] [-DefaultCordaRoot configrootdir]
[-Config instanceconfig]
```

Table 3 lists and describes the command line options.

Table 3. Javahost Service Command Line Options

Options	Operating Systems	Description
-regserver	Windows	Registers the Siebel Analytics Javahost service in manual startup mode.
-regserverauto	Windows	Registers the Siebel Analytics Javahost service in automatic startup mode.

Table 3. Javahost Service Command Line Options

Options	Operating Systems	Description
-unregserver	Windows	Unregisters the Siebel Analytics Javahost service.
-service	UNIX, Windows	Executes Javahost in Service mode. On Windows, this parameter should never be used explicitly. Instead, use the control panel Services applet to start and stop the Siebel Analytics Javahost service.
-V	Windows	Displays the version.
-SawConfigRoot configrootdir	UNIX, Windows	Specifies the location of the config directory. The default location is {SiebelAnalyticsData}/web/config.
-DefaultCordaRoot configrootdir	UNIX, Windows	Specifies the location of the Corda installation root directory. This parameter is considered by the Javahost service only if the configuration key JavaHost/Charts/CordaRoot is not set in instanceconfig.xml.
-Config instanceconfig	UNIX, Windows	Provides the path to instanceconfig.xml. The default path is {SiebelAnalyticsData}/web/config/ instanceconfig.xml
-Port port	UNIX	Identifies the Javahost listening port.
-Host hostname	UNIX	Identifies the computer running the Javahost service.

In UNIX, the shutdown.sh parameters serves one purpose, which is to pass connection information (host and port) so that a shutdown signal can be sent to the Javahost service. The following rules explain how these parameters interact:

- If either the -Host or the -Port parameter is specified, then shutdown.sh ignores the -Config and the -SawConfigRoot parameters.
- If -Host is specified and -Port is not, then shutdown.sh uses 9810 as the port number.
- If the -Config parameter is set, then shutdown.sh ignores the -SawConfigRoot parameter and instead uses the instanceconfig.xml file to find the Javahost listening port. It expects that the Javahost service is running on the local computer (Host=localhost).
- In the absence of all other parameters, shutdown.sh uses the -SawConfigRoot parameter to find the location of instanceconfig.xml file.
- If no parameters are specified, then shutdown.sh uses Host=localhost and port=9810.

Configuring the Javahost Service

To configure the Javahost service, edit the Javahost service configuration parameters in the instanceconfig.xml file. Table 4 lists and describes the Javahost parameters. The parameters are identified by their relative path starting from the /WebConfig/ServerInstance node.

Table 4. Javahost Service Configuration Parameters

Javahost Subcomponents	Parameter	Parameter Type	Description
Batik	JavaHost/Batik/ InputStreamLimitInKB	Integer, positive values only	Maximum input size for Batik renderer requests. A value of zero deactivates this limit. Default: 1024
Corda	JavaHost/Charts/ ChartRoot	String	Path to the Corda chart_root directory. Default: {CordaRoot}/chart_root.
	JavaHost/Charts/ CordaRoot	String	Path to the Corda installation. Default: Value passed in the command line.
	JavaHost/Charts/ EnableConsoleOutput	Yes/No	Enable or disable the Corda console diagnostic messages. Default: No
	JavaHost/Charts/ InputStreamLimitInKB	Integer, positive values only	Maximum input size for charts requests. A value of zero deactivates this limit. Default: 1024
FOP processor	JavaHost/PDF/ InputStreamLimitInKB	Integer, positive values only	Maximum input size for PDF requests. A value of zero deactivates this limit. Default: 1024
	JavaHost/PDF/ UserConfigFile	String	FOP configuration file. Default: {SADATADIR}/web/config/ userconfig.xml
Socket	JavaHost/Listener/ PermittedClientList	String	Comma-separated list of IP addresses and host names from which Javahost accepts incoming connections. To accept all client connections, set this parameter to an asterisk (*). Default: 127.0.0.1
	JavaHost/Listener/Port	Integer, positive values only	TCP/IP listening port Default: 9810

Table 4. Javahost Service Configuration Parameters

Javahost Subcomponents	Parameter	Parameter Type	Description
Tuning	JavaHost/JobManager/ IdleThreadTimeoutMls	Integer, positive values only	Idle timeout (in seconds) for a thread in the thread pool. After timeout expires the thread is shut down. Default: 30000 (5 minutes)
	JavaHost/JobManager/ MaxPendingJobs	Integer, positive values only	Maximum number of pending process requests after which Javahost starts to reject them. Default: 100
	JavaHost/JobManager/ MaxThreads	Integer, positive values only	Maximum number of threads for the internal thread pool. Default: 100
	JavaHost/JobManager/ MinThreads	Integer, positive values only	Minimum number of threads for the internal thread pool. Default: 1
	JavaHost/ MessageProcessor/ SocketTimeout	Integer, positive values only	Idle timeout (in milliseconds) for socket after which socket is returned to the "idle" sockets pool. Javahost uses a socket polling mechanism to wait for new data on the whole set of "idle" sockets in a single thread. Default: 5000 (5 seconds)
UNIX JVM	JVM parameters (Unix)	N/A	In UNIX the Java command line parameters can be modified by changing the value of the JAVAOPTIONS variable in the {SAROOT}/web/sawjvahost/setup/ saw.sh script.

Table 4. Javahost Service Configuration Parameters

Javahost Subcomponents	Parameter	Parameter Type	Description
Windows JVM	JavaHome	String	Path to the root directory of JDK or JRE installation. Default: Value of JAVA_HOME environment variable
	JavaHost/InitLoggerDir	String	Absolute path to the directory where sawjavahostsvc.exe writes log information before it loads java. If the value of this parameter is empty. logging is disabled. Default: {SADATADIR}/web/log/javahost, where {SADATADIR} is the full path to SiebelAnalyticsData directory.
	JavaHost/JniLibrary	String	Absolute path to jvm.dll. Default: {JavaHome}/jre/bin/server/jvm.dll If that file does not exist, then: {JavaHome}/bin/server/jvm.dll, where {JavaHome} is the path to JDK or JRE installation.
	JavaHost/JVMOptions	String	Java command line parameters. Default: -Xms128m -Xmx256m -Xrs "-Djava.class.path={CLASSPATH}" "-Djava.awt.headless=true" "-Djava.util.logging.config.file={SADATADIR}/web/config/logconfig.txt" where {CLASSPATH} is the list of javahost jar files separated by semicolons.
	JavaHost/UseDefaultJVMOptions	Yes/No	When set to No, JavaHost/JVMOptions specifies exact command line for Java. When set to Yes, the value of JVMOptions parameter is merged with its default value. Default: Yes

Javahost Service Logging

The Javahost service uses a standard Java logging engine. By default, Javahost uses a logging configuration file located at {SADATADIR}/web/config/logconfig.txt. For the logging file format description, see publicly available Java documentation.

On Windows, several initialization messages could be written before Java is initialized. Use the JavaHost/InitLoggerDir configuration key to control the location where those messages are written.

3 Administering Siebel Answers

This chapter describes procedures that are used to administer Siebel Answers. For an introduction to Siebel Answers, see *Siebel Analytics User Guide*.

This chapter contains the following topics:

Managing Settings for the Siebel Analytics Web Chart Image Server

Siebel Analytics Web uses a third-party charting engine to render charts (PopChart Image Server from CORDA Technologies, Inc.). You can override the default image type and browser client connection type by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entries:

You need to create the tags <POP> and </POP> after the <ServerInstance> tag, and place your entries between the <POP> and </POP> tags.

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Specifying the Siebel Analytics Chart Image Type

You can override the image type generated by the PopChart Image Server. The internal default is Flash.

The other choices are SVG (W3C Scalable Vector Graphics), PNG (W3C Portable Network Graphics), and JPEG. Flash and SVG images provide the greatest degree of interaction because they support mouseover behaviors (such as pop-up data labels), navigation, and drilling.

The following entry is an example:

```
<POP>
   <DefaultImageType>PNG</DefaultImageType>
</POP>
```

About Downloading and Using Flash as the Default Image Type in Siebel Analytics

In some organizations, users are instructed to download the latest Flash software from a corporate location instead of the vendor's Web site. The default download source for Siebel Analytics Web is the vendor's Web site. You can modify the default download source to point to another location. Then, when users view a chart in Siebel Analytics and a newer version of the Flash software is available on the corporate server, they can be prompted to download the newer version. This section contains the following topics:

- "Modifying the Default Flash Download Source" on page 34
- "Enabling a Download Prompt for New Flash Versions" on page 34

Modifying the Default Flash Download Source

The default download source for the Flash plug-in is the vendor's Web site. You can change this to another location by modifying the Siebel Analytics Web configuration file instanceconfig.xml to point to the location that holds the Flash code base.

NOTE: You need to create the tags <FlashCodeBase> and </FlashCodeBase> after the <Charts> tag, and place your entries between the <FlashCodeBase> and </FlashCodeBase> tags.

The following entry is an example:

```
<Charts>
   <FlashCodeBase>\\CORPORATE\Download\Flash</FlashCodeBase>
</Charts>
```

Enabling a Download Prompt for New Flash Versions

After modifying the default Flash download source, you can enable a download prompt by creating a new classID for the Flash ActiveX control to add a custom global identifier (clsid) property. You can obtain the current global identifier property from any machine where Siebel Analytics Web charting is being used. (For version 7.7.1, the global identifier property used by Siebel Analytics is D27CDB6E-AE6D-11CF-96B8-444553540000.) The custom global identifier property must contain the same number of characters and dashes as the global identifier used in the default Flash ActiveX control.

The following entry is an example:

```
<Charts>
   <FlashCLSID>E38CDB6E-BA6D-21CF-96B8-432553540000</FlashCLSID>
</Charts>
```

You should test flash charts independent of Siebel Analytics to make sure that they function with the custom global identifier property.

Specifying How the Siebel Analytics Web Server Connects to the PopChart Image Server

You can override how the Siebel Analytics Web server connects to the PopChart Image Server. The default is http://AnalyticsWeb_machine:2001/?, where AnalyticsWeb_machine is the machine name where Siebel Analytics Web is running, and 2001 is the port number.

This setting must be fully qualified with the port number (if other than 80) using the notation as shown.

The following entry is an example:

```
<POP>
   <ServerPrefix>http://AnalyticsWeb_machine:85/?</ServerPrefix>
</POP>
```

Managing the Siebel Analytics Web Charting Settings

You can change certain Siebel Analytics Web charting settings from their internal default settings by adding entries to the Siebel Analytics Web configuration file instanceconfig.xml:

- “Specifying the Location of Temporary Storage for Chart Cache in Siebel Analytics Web” on page 35
- “Specifying the Interactive Behavior of Charts in Siebel Analytics Web” on page 36
- “Specifying the URL for Chart Navigation in Siebel Analytics Web” on page 36

You need to create the tags <Chart> and </Chart> after the <ServerInstance> tag, and place your entries between the <Chart> and </Chart> tags.

For information about working in the configuration file instanceconfig.xml, see “Making Siebel Analytics Web Configuration Changes” on page 11.

Specifying the Location of Temporary Storage for Chart Cache in Siebel Analytics Web

You can override the internal default entry for the location used by Siebel Analytics Web as temporary storage for chart cache. The internal default is dirletter:\SiebelAnalyticsData\Temp\nQWCharts, where dirletter is the drive where the Siebel Analytics Web software is installed.

The location should be a fully-qualified path name, such as d:\SiebelWebChartCache. The folder or directory structure must exist before you start the Analytics Web Server service again.

The following entry is an example:

```
<Charts>
   <CacheDirectory>/usr/local/SiebelAnalytics/Data/temp/chartcache</CacheDirectory>
</Charts>
```

Specifying the Interactive Behavior of Charts in Siebel Analytics Web

You can override the default interactive behavior of Siebel Analytics Web charts. Valid entries are Drill, Navigate and None. The default is Drill. This means that charts are created drillable by default.

If you specify Navigate, charts navigate to the URL described in "Specifying the URL for Chart Navigation in Siebel Analytics Web" on page 36. If you specify None, charts are not interactive; for example, clicking on a chart or chart region does nothing.

The following entry is an example:

```
<Charts>
   <DefaultInteraction>Navigate</DefaultInteraction>
</Charts>
```

Specifying the URL for Chart Navigation in Siebel Analytics Web

If you specify Navigate as the interactive behavior of charts (described in "Specifying the Interactive Behavior of Charts in Siebel Analytics Web" on page 36), you can override the default URL to which charts navigate. The internal default URL is http://www.siebel.com/.

If the interactive behavior is not Navigate, this entry is ignored.

The following entry is an example:

```
<Charts>
   <DefaultNavigationPath>http://www.intranet.com/</DefaultNavigationPath>
</Charts>
```

Configuring Siebel Answers Pivot Table Settings

You can change certain pivot table settings from their internal defaults by adding entries to the Siebel Analytics Web configuration file instanceconfig.xml:

- "Specifying the Maximum Number of Records to Process in a Siebel Analytics Pivot Table"
- "Specifying the Maximum Number of Populated Cells in a Siebel Analytics Pivot Table"

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Specifying the Maximum Number of Records to Process in a Siebel Analytics Pivot Table

You can override the maximum number of records that can be processed by a pivot table. The internal default is 20000.

The following entry is an example:

```
<CubeMaxRecords>30000</CubeMaxRecords>
```

Specifying the Maximum Number of Populated Cells in a Siebel Analytics Pivot Table

You can override the maximum number of populated cells that Siebel Analytics Web allows in a pivot table. The internal default is 150000. If the user exceeds this value, the server returns an error message when the pivot table is rendered.

The following entry is an example:

```
<CubeMaxPopulatedCells>160000</CubeMaxPopulatedCells>
```

Configuring the Maximum Number of Rows in a Siebel Answers Table View

You can override the maximum number of rows that can appear in a Table view by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. The internal default is 10000. If the user exceeds this value, the server returns an error message when the table view is rendered.

NOTE: This entry applies to the Table view, not the Pivot Table view.

The following entry is an example:

```
<ResultRowLimit>12000</ResultRowLimit>
```

Adding Support for Navigation and Drill Down in Siebel Answers

When you create a request, you can allow users to navigate to related requests and content. If the Siebel Analytics Server administrator set up dimensions and dimensional hierarchies for the subject area, users can drill down on data results presented in charts, tables and pivot tables to obtain more detailed information.

There are no specific privilege settings that impact access to Siebel Analytics Web Navigation and drill down features, which are available to all users.

Dimensional hierarchies are a system of levels in a dimension that are related to each other by one-to-many relationships. For example, a region hierarchy might be defined as a postal code rolling up to a city, rolling up to a region. There is exactly one city and one region corresponding to a single postal code, but there might be many postal codes corresponding to a single city, and there might be many cities corresponding to a single region.

Drill down allows users to move through the dimensional hierarchies associated with the subject area to obtain more detailed results. For example, if the results contain a total sales column, the user might be able to click total sales to drill down for sales by region, and then click a region to see sales by city in that region. The information available to Siebel Analytics Web users through drill down is constrained by the dimensional hierarchies configured by the Siebel Analytics Server administrator for the specific subject area.

For information about setting up dimensions and hierarchies, see *Siebel Analytics Server Administration Guide*.

Changing the Default Currency in Siebel Answers

You can change the default currency displayed in the Siebel Answers user interface, for example, from French Francs to Euros.

If you are using Siebel Analytics applications, with no customizations, you only need to set the default data warehouse currency.

If you have created additional subject areas, the currency column data is in the number format, and you need to specify the currency for the customized subject area as described in the second procedure.

For more information about using formatting functions in Siebel Answers, see *Siebel Analytics User Guide*.

To set the default data warehouse currency

1. Open the currencies.xml file in the directory \\SiebelAnalyticsData\Web\Config.
2. Search for the currency that you want to make the default, for example USD, CAD, PEN, or MAD.
3. Copy the entire currency element.

   For example, copy the currency tag for the Euro:

   ```
   - <Currency tag="int:euro-l" type="international" symbol="_"
   displayMessage="kmsgCurrencyEuroLeft" digits="2" format="$ #">

   <negative tag="minus" format="-$ #" />

   </Currency>
   ```

4. Search for the text string int:wrhs, located towards the top of the file.
5. Select the entire element and replace it by pasting the copied element over it.
6. Replace the tag attribute so it reads int:wrhs.

   For example, replace `tag="int:euro-l"` with `tag="int:wrhs"`.

7. Restart the Analytics Web Server service.

To specify the currency for customized subject areas

1. In Siebel Answers, modify the request that uses the subject area.
2. Click the Format Column button for the currency column.

   The Column Properties dialog box appears.

3 At the Value Format tab, in the Data Format area, click the following option:

   Override Default Data Format

4 In the Treat Numbers As drop-down list, select Currency.

5 In the Currency Symbol drop-down list, select the currency symbol.

6 To save this as the system-wide default for this data type, click the Save button and then select the appropriate option.

7 Click OK when you are done, and then repeat the preceding steps for any other columns you want to change.

Nesting Folders in the Selection Pane in Siebel Answers

To make selections easy for users to discern in the selection pane, you can set up the presentation layer in the Analytics Server Administration Tool to give the appearance of nested folders. For example, you can make the Sales Facts folder appear as a subfolder in the Facts folder. You can accomplish this during the construction of the presentation layer by prefixing the name of a folder to appear as a subfolder with a hyphen (-) and a space.

The preconfigured Siebel Analytics repository for Siebel operational applications provides examples of this construction. For more information, see *Siebel Analytics Server Administration Guide*.

Blocking Requests in Siebel Answers

Administrators may want to block specific requests, such as requiring users to include certain columns with others, or requiring filters when certain columns are requested. Siebel Answers includes an API that you can use to block queries based on the criteria specified in the user's request, or based on formulas in the request. Administrators can access the API using JavaScript to check conditions and validate requests.

This section contains the following topics:

Blocking Requests Based on Criteria

When a user attempts to execute a request that your code blocks, you can display an error message, and the request will not be executed. The answerstemplates.xml file includes a message named kuiCriteriaBlockingScript that can be overridden to either define or include JavaScript that defines a validateAnalysisCriteria function. By default, this message contains a function that always returns True. It should be overridden using the procedures described in "Customizing the Siebel Analytics Web User Interface Using XML Message Files" on page 196.

Siebel Answers calls your validateAnalysisCriteria function when the user tries to execute the request. The function can return True if the request is not blocked, or False or a message if the request is blocked. If a message or a value other than False is returned, the message is displayed in a popup window. In either case, the query is blocked.

The following code example shows the blocking of a query.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="QueryBlocking" table="Messages">

   <WebMessage name="kuiCriteriaBlockingScript" translate="no">
      <HTML>
         <script language="javascript" src="fmap:myblocking.js" />
      </HTML>
   </WebMessage>

   </WebMessageTable>
</WebMessageTables>

Sample blocking script in …/SiebelAnalyticsData/web/Res/myblocking.js

// This is a blocking function. It makes sure users pick what I want them to.
function validateAnalysisCriteria(analysisXml)
{

   // Create the helper object
   var tValidator = new CriteriaValidator(analysisXml);

   // Validation Logic
   if (tValidator.getSubjectArea() != "Paint")
      return "Why don't you try Paint?";

   if (!tValidator.dependentColumnExists("Markets","Region","Markets","District"))
   {

      // If validation script notifies user, then return false
      alert("Region and District go so well together, don't you think?");
      return false;
   }

   if (!tValidator.dependentColumnExists("Sales Measures","","Periods","Year"))
   return "You picked a measure so pick Year!";

   if (!tValidator.filterExists("Sales Measures","Dollars"))
   return "Why don't you filter on Dollars?";

   if (!tValidator.dependentFilterExists("Markets","Market","Markets"))
   return "Since you're showing specific Markets, please filter the markets.";

   var n = tValidator.filterCount("Markets","Region");
   if ((n <= 0) || (n > 3))
      return "Please select 3 or fewer specific Regions";

   return true;
}
```

If you do not override the function using the template as described previously, or if the function returns anything other than False, the criteria is considered to be valid and the request is issued. The criteria is validated using this same mechanism for preview and save operations as well.

Blocking Requests Based on Formula

Siebel Answers provides a hook that allows you to create and incorporate a JavaScript validation function that is called from Siebel Answers when a user enters or modifies a column formula. If the call fails and returns a message, Siebel Answers displays the message and cancels the operation. Additionally, helper functions are available so the query blocking function can check for filters, columns, and so on, rather than traversing the DOM manually. For more information on the helper functions, see "Validation Helper Functions" on page 42.

The criteriatemplates.xml file includes a message named kuiFormulaBlockingScript that can be overridden to include JavaScript that defines a validateAnalysisFormula function. By default, this message contains a function that always returns True.

Siebel Answers calls validateAnalysisFormula before applying changes made by the user. If the function returns True, the formula is accepted. If the function returns False, the formula is rejected. Otherwise, the return value from the function is displayed in the message area beneath the formula, as it does currently when an invalid formula is entered.

The user has the option to click OK to ignore the error. To display your own alert and allow the user to continue, your function should return True. To block the query, return False or a message. Your function should investigate the formula passed to it using JavaScript string and regular expression techniques for validation.

The following code example shows a sample custom message.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="QueryBlocking" table="Messages">

      <WebMessage name="kuiFormulaBlockingScript" translate="no">
         <HTML>
            <script language="javascript" src="fmap:myblocking.js" />
         </HTML>
      </WebMessage>

   </WebMessageTable>
</WebMessageTables>
```

The following code example shows blocking based on the formula entered.

```
// This is a formula blocking function. It makes sure the user does not enter an
unacceptable formula.
function validateAnalysisFormula(sFormula, sAggRule)
{

   // we don't allow the use of concat || in our formulas
   var concatRe = /\|\|/gi;
   var nConcat = sFormula.search(concatRe);
```

```
   if (nConcat >= 0)
      return "You used concatenation (character position " + nConcat + "). That is
not allowed.";

   // no case statements please
   var caseRe = /CASE.+END/gi;
   if (sFormula.search(caseRe) >= 0)
      return "Please do not use a case statement.";

   // Check for a function syntax: aggrule(formula) aggrule should not contain a '.'
   var castRe = /^\s*\w+\s*\(.+\)\s*$/gi;
   if (sFormula.search(castRe) >= 0)
      return "Please don't use a function syntax such as RANK() or SUM().";

   return true;
}
```

Validation Helper Functions

These functions are defined within a JavaScript file named answers/queryblocking.js. Table 5 contains the list of helper functions and their descriptions.

Table 5. Validation Helper Functions

Validation Helper Function	Description
CriteriaValidator.getSubjectArea()	Returns the name of the subject area referenced by the request. It generally is used in a switch statement within the function before doing other validation. If the request is a set-based criteria, it returns null.
CriteriaValidator.tableExists (sTable)	Returns True if the specified table has been added to the request by the user, and False if the table was not added.
CriteriaValidator.columnExists (sTable, sColumn)	Returns True if the specified column has been added to the request by the user, and False if the column was not added.
CriteriaValidator.dependentColumn Exists(sCheckTable, sCheckColumn, sDependentTable, sDependentColumn)	Checks to make sure that the dependentColumn exists if the checkColumn is present. It returns True if either the checkColumn is not present, or the checkColumn and the dependent column are present. If checkColumn and dependentColumn are null, the tables are validated. If any column from checkTable is present, a column from dependentTable must be present.
CriteriaValidator.filterExists(sFilter Table, sFilterColumn)	Returns True if a filter exists on the specified column, and False if no filter is present.

Table 5. Validation Helper Functions

Validation Helper Function	Description
CriteriaValidator.dependentFilterExists(sCheckTable, sCheckColumn, sFilterTable, sFilterColumn)	Checks to make sure that the dependentFilter exists if the checkColumn is present in the projection list. It returns True if either the checkColumn is not present, or the checkColumn and the dependent filter are present.
CriteriaValidator.filterCount(sFilterTable, sFilterColumn)	Returns the number of filter values specified for given logical column. If the filter value is "equals," "null," "notNull," or "in," it returns the number of values chosen. If the column is not used in a filter, it returns zero. If the column is prompted with no default, it returns -1. For all other filter operators (such as "greater than," "begins with," and so on) it returns 999, because the number of values cannot be determined.

Specifying View Defaults for Siebel Answers and Siebel Intelligence Dashboard Users

You can control some aspects of the initial state of new views that are added to a request within Siebel Answers and new objects that are added to a dashboard page. You do this by customizing the appropriate XML message files to override the default values distributed with Siebel Analytics Web.

Some of the default values you can customize are shown in the following list:

- Allowing the sorting of table columns in a dashboard.
- Adding a default page footer to new reports.
- Preventing the auto-previewing of results when working with a view.
- Specifying which views are contained in the compound layout view.
- Specifying which links to display with an embedded report in the dashboard.
- Allowing newly created dashboard sections to be collapsible.

XML Message Files for View Defaults

This section describes the XML message files to customize to override the view defaults distributed with Siebel Analytics Web.

NOTE: For information about the core tasks required to customize XML message files, see "Customizing the Siebel Analytics Web User Interface Using XML Message Files" on page 196.

For Siebel Answers, the file answerstemplates.xml includes a message named kuiCriteriaDefaultViewElementsWrapper from within kuiAnswersReportPageEditorHead. This message includes two additional messages, kuiCriteriaDefaultViewElements, in which you can define default values, and kuiCriteriaDefaultViewElementsMask, in which masks are defined.

NOTE: The mask XML message is protected and you cannot modify its contents.

The wrapper message adds the combined XML into a JavaScript variable, kuiDefaultViewElementsXML, that is used to apply the new default values.

For Siebel Intelligence Dashboards, the file dashboardtemplates.xml includes a message named kuiDashboardDefaultElementsWrapper that adds XML into a JavaScript variable named kuiDefaultDashboardElementsXML for use within the dashboard editor.

Examples of Customizing Siebel Answers and Siebel Intelligence Dashboard Default Values

The following sections provide examples of customizing default values:

NOTE: For information about the core tasks required to customize XML message files, see "Customizing the Siebel Analytics Web User Interface Using XML Message Files" on page 196. The examples in this section assume you have read this information.

Adding a Default Header or Footer to New Reports

You can specify to the system that default headers and footers appear on all new reports. Footers, for example, can contain messages such as a confidentiality notice, the company's name, and so on. You can specify a default header or footer by creating an XML message that specifies the text and formatting that should be applied, and then deploying it to the Siebel Analytics Web Server.

The following XML code example creates a footer that contains the text "Acme Confidential" in bolded, red letters.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">

<WebMessage name="kuiCriteriaDefaultViewElements" translate="no"><HTML>
   <view signature="compoundView" >
      <pageProps pageSize="a4">
```

```
         <pageFooter showOnDashboard="true" show="true">
            <zone type="top"><caption>[b]Acme Confidential[/b]</caption>
            <displayFormat fontColor="#FF0000"/></zone>
         </pageFooter>
      </pageProps>
   </view>
</HTML></WebMessage>

   </WebMessageTable>
</WebMessageTables>
```

Allowing Sorting in Tables in Siebel Intelligence Dashboards

By default, Siebel Analytics table views are not sortable within dashboards and result views. To make tables sortable, create an XML message that specifies the text and formatting that should be applied, and deploy it to the Siebel Analytics Web Server. Then, when a new table view is created in Siebel Answers, the option to allow sorting in dashboards will be selected by default.

The following XML code example turns on the option to allow sorting.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">

<WebMessage name="kuiCriteriaDefaultViewElements" translate="no"><HTML>
   <view signature="tableView" sortable="true" />
</HTML></WebMessage>

   </WebMessageTable>
</WebMessageTables>
```

Preventing Auto-Previewing of Results in Siebel Answers

Siebel Analytics displays the results of the request when editing most views within Answers. If you prefer that the user explicitly ask to view the results, you can create an XML message that specifies that auto-preview should be disabled when new views are created. The user can still click the display results link to view the results when editing a view.

The following XML code example disallows the auto-previewing of results when working with a view in Siebel Answers.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">

<WebMessage name="kuiCriteriaDefaultViewElements" translate="no"><HTML>
   <view signature="tableView" showToolBar="true" showHeading="true" />
   <view signature="pivotTableView" autoPreview="false" />
   <view signature="titleView" autoPreview="false" />
   <view signature="viewSelector" autoPreview="false" />
   <view signature="htmlviewnarrativeView" autoPreview="false" />
   <view signature="tickerview" autoPreview="false" />
   <view signature="htmlview" autoPreview="false" />
</HTML></WebMessage>
```

```
   </WebMessageTable>
</WebMessageTables>
```

Setting Defaults for the Compound Layout View in Siebel Answers

Siebel Answers displays the results of a newly formed request as a title view followed by a table view. You can create an XML message that specifies that the compound view should default to a different collection of views, such as a table view followed by a filters view. The user can still add and rearrange views within the compound layout view.

The following XML code example sets the default compound layout view to a table view followed by a filters view.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">

<WebMessage name="kuiCriteriaDefaultViewElements" translate="no"><HTML>
   <view signature="compoundView" >
      <cv signature="tableView" />
      <cv signature="filtersView" />
   </view>
</HTML></WebMessage>

   </WebMessageTable>
</WebMessageTables>
```

Changing Siebel Intelligence Dashboard Section Defaults

By default, Siebel Analytics displays the results of drilling in the dashboard on a new page, does not show section names in the dashboard, and does allow users to expand and collapse sections. You can change these default values by creating an XML message that specifies that new default values for the dashboard section.

The following XML code example makes section heads visible, enables drilling, and does not allow users to collapse the sections.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">

<WebMessage name="kuiDashboardDefaultElements" translate="no"><HTML>
   <element signature="dashboardSection" drillInline="true" showHeading="true"
collapsible="false" />
</HTML></WebMessage>

   </WebMessageTable>
</WebMessageTables>
```

Including Refresh and Modify Links with Reports on Siebel Dashboards

By default, Siebel Analytics displays the results of embedded reports within the dashboard without including any links. If you prefer that newly added reports default to having Modify and Refresh links, for example, you can create an XML message that specifies that the report elements should behave this way. A user editing the dashboard can still modify this behavior using the menus within the dashboard editor.

In the XML message file, the links attribute can contain any combination of the letters mfrdg to add the indicated link, as shown in Table 6.

Table 6. Attribute Values for Adding Links to Embedded Reports

Attribute	Link Added to Report on the Dashboard
m	Modify
f	Print
r	Refresh
d	Download
g	Add to Briefing Book

The following XML code example adds Modify and Refresh links to new reports embedded in dashboards.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">

<WebMessage name="kuiDashboardDefaultElements" translate="no"><HTML>
   <element signature="reportView" display="embed" links="mr" />
</HTML></WebMessage>

   </WebMessageTable>
</WebMessageTables>
```

Specifying Dashboard Page Defaults Including Headers and Footers

By default, Siebel Analytics prints dashboards without headers or footers, and in a portrait orientation. If you prefer that newly added dashboard pages default to having a custom header and footer and print in landscape orientation, you can create an XML message that specifies these items. A user editing the dashboard can still modify this behavior using the menus within the dashboard editor.

The following XML code example adds a custom header and footer to a dashboard page and specifies landscape orientation.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables xmlns:sawm="com.siebel.analytics.web.messageSystem">
   <WebMessageTable system="Answers" table="ViewDefaults">
```

```
<WebMessage name="kuiDashboardDefaultElements" translate="no"><HTML>

<element signature="dashboardPage" personalSelections="false">
     <pageProps orientation="portrait" printRows="all" pageSize="a4">
         <pageHeader showOnDashboard="true" show="true">
            <zone type="top"><caption>[b]Acme is Cool[/b]</caption>
            <displayFormat fontSize="9pt" hAlign="center"
fontColor="#FFFFFF" backgroundColor="#000000"/></zone>
         </pageHeader>
         <pageFooter showOnDashboard="true" show="true">
            <zone type="top"><caption>[b]CONFIDENTIAL
@{timeCreated[mm/dd/yy]}[/b]</caption>
            <displayFormat fontSize="7.5pt" hAlign="center"
fontColor="#999999" borderColor="#CC99CC" fontStyle="italic"
borderPosition="all" borderStyle="single"/></zone>
         </pageFooter>
       </pageProps>
   </element>

</HTML></WebMessage>

   </WebMessageTable>
```

4 Administering Siebel Delivers

This chapter describes procedures used to administer Siebel Delivers. For information about using Siebel Delivers, see *Siebel Analytics User Guide*.

This chapter contains the following topics:

About Siebel Delivers iBots and Impersonation

Siebel Delivers makes use of intelligence agents or Bots (called iBots). iBots are software-based agents driven by schedule or events that access, filter, and perform analytics on data based upon defined criteria. Users receive information from iBots in the form of alerts that appear on their designated delivery devices or dashboards.

To create an iBot, Siebel Analytics Web administrators and users use Siebel Delivers to define the actions the iBot is to perform. Siebel Analytics Web gathers information about the priority, delivery devices, user, and other characteristics; packages that information into a job; and tells Siebel Analytics Scheduler when it wants the job to execute.

Because the Scheduler runs these jobs on behalf of users without accessing or storing their passwords, Siebel Analytics Server allows the Scheduler and Siebel Analytics Web to impersonate users. This is done by configuring the Scheduler to use a user ID and password with administrator privileges that can act on behalf of other users. An iBot logs on to the system using this user ID and password, and then the Scheduler executes the job on behalf of the user.

NOTE: If Siebel Analytics Server is configured to authenticate users through database logons, then impersonation is not allowed. Siebel Delivers works with database authentication provided that only the initialization block set up for authentication in the Siebel Analytics Server Administration Tool uses a connection pool with pass-through authentication. That connection pool cannot be used for any other initialization block or request.

For information about user authentication options, see *Siebel Analytics Server Administration Guide*. For information about the Scheduler, see *Siebel Analytics Scheduler Guide*.

About Siebel Delivers iBots and AntiVirus Software

Some antivirus software programs, such as Norton AntiVirus, enable a script-blocking feature, which tries to block all calls made by scripts to system objects (such as the Windows file system object) that the antivirus software deems unsafe.

If you launch a script as part of post-iBot processing, this antivirus feature may cause unexpected results. If you are running antivirus software with a script-blocking feature on the machine where Siebel Analytics Scheduler is installed, you should disable the script-blocking feature to prevent the software from unexpectedly blocking iBot script calls.

Viewing Entries in the Siebel Delivers iBot Log Directory

If an iBot fails to execute fully or if debugging is turned on in Siebel Analytics Scheduler, a log file is generated for the iBot.

The location for iBot log files is specified on the iBots tab of the Job Manager Configuration dialog box in Siebel Analytics Scheduler. The default location is the Log directory in the Siebel Analytics installation directory on the machine where Siebel Analytics Scheduler is installed.

The log file name has the following format:

NQiBot-JobID-InstanceID.xxx

where:

NQibot	The preface for all iBot log files.
JobID	The Scheduler job ID for the iBot.
InstanceID	The Scheduler instance ID for the iBot.

xxx The file extension:

- .err for iBot error log files.
- .log for debug log files.

The iBot error and debug log files are written as separate files for each iBot instance that fails to execute. You can use a text editor to view the files. Entries are generally self-explanatory. Exit codes are generic and do not indicate any particular condition.

The presence of an error log does not necessarily mean that an iBot failed completely. For example, suppose an iBot delivers content to multiple email addresses. If some of the addresses are invalid or the mail server is down, an error log is generated for the iBot.

For more information about the Scheduler, see *Siebel Analytics Scheduler Guide*.

Disabling Siebel Delivers

Siebel Delivers is an optional component of Siebel Analytics Web that is enabled by default for organizations that have purchased the appropriate license. To disable Siebel Delivers, you can edit the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. The default value is Y (Siebel Delivers is enabled). To disable Siebel Delivers, set this to N.

You need to create the tags <Alerts> and </Alerts> after the <ServerInstance> tag, and place your entries between the <Alerts> and </Alerts> tags.

The following entry is an example:

```
<Alerts>
   <Enabled>No</Enabled>
</Alerts>
```

For information about working in the configuration file instanceconfig.xml, see “Making Siebel Analytics Web Configuration Changes” on page 11.

Specifying the Machine Running Siebel Analytics Scheduler

You can identify the machine running Siebel Analytics Scheduler by editing the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. If Siebel Analytics Scheduler is running on the local machine, this entry is populated with the name of the local machine.

You need to create the tags <Alerts> and </Alerts> after the <ServerInstance> tag, and place your entries between the <Alerts> and </Alerts> tags.

The following entry is an example:

```
<Alerts>
   <ScheduleServer>Server02</ScheduleServer>
</Alerts>
```

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

NOTE: If you are migrating a Siebel Analytics environment to a new system, make sure you also migrate the Siebel Analytics Server repository file and the Scheduler tables. The Scheduler tables are required for iBots.

Changing the Directory in Which Siebel Delivers iBot Deliveries Are Stored

You can specify the directory in which iBot deliveries are stored by editing the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. The deliveries directory is, by default, stored in the same location as the Web Catalog. (The value described in "Changing the Name and Location of the Siebel Analytics Web Catalog" on page 69 defines where the Web Catalog is stored.) At startup, Siebel Analytics Web attempts to create the deliveries directory.

You need to create the tags <Web> and </Web> after the <ServerInstance> tag, and place your entries between the <Web> and </Web> tags.

The following entry is an example:

```
<PersistentStorageDirectory>/usr/local/SiebelAnalytics/Data/web/catalog/TestDelivery</
PersistentStorageDirectory>
```

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Integrating Siebel Delivers with Siebel Workflow

You can use the Advanced tab in Siebel Delivers to set up iBots to trigger workflows in the Siebel Workflow application. The procedures to configure this capability are given in *Siebel Analytics Scheduler Guide*.

By default, only Siebel Analytics Web administrators have the necessary privileges to set up iBots to trigger workflows.

For information about the Advanced tab in Siebel Delivers, see *Siebel Analytics User Guide*.

Using Siebel Delivers to Seed the Siebel Analytics Server Cache

You can use the Destinations tab in Siebel Delivers to set up iBots to seed the Siebel Analytics Server cache. Seeding the cache can improve response times for users when they run requests in Siebel Answers or view requests that are embedded on their dashboards. Do this by scheduling iBots to execute requests that refresh this data.

For more information about Siebel Analytics Server cache, see *Siebel Analytics Server Administration Guide*.

For information about the Destinations tab in Siebel Delivers, see *Siebel Analytics User Guide*.

About Permission Settings for Siebel Delivers and iBots

The permission settings for Siebel Delivers and iBots are available in the Siebel Delivers section on the Privilege Administration page in Siebel Analytics Administration.

When users are granted access to the Publish for Subscription privilege, they also need to have Change/Delete permission to the shared iBots object and child objects in the Siebel Analytics Web Catalog.

You can grant the Change/Delete permission by using the Manage Catalog feature. For information about using the Manage Catalog feature, see "Administering Items in the Siebel Analytics Web Catalog" on page 92.

About the SA System Subject Area and Scheduling iBot Deliveries

This section applies only to organizations using Siebel Analytics operational applications.

When the SA System subject area is being used, iBot deliveries cannot be made to user IDs that are defined only in the Siebel Analytics repository. Such internally defined user IDs include the Administrator user ID.

The SA System subject must return rows for any acceptable user of Siebel Analytics. In all Siebel Analytics operational applications, the Administrator user ID is not defined in the Siebel OLTP, and therefore is not returned from queries to the SA System subject area. You should test alerts with a valid Siebel OLTP user.

Viewing Information About Active Siebel Delivers iBot Sessions

You can view the following information about currently active iBot sessions triggered by the Scheduler:

- A list of active iBots per session.
- The recipients for each active iBot.

The capability to view information about active iBot sessions is available to users who have the following privilege set in the Admin: General section of the Privilege Administration screen:

Manage iBot Sessions

This privilege is typically granted to users defined as Siebel Analytics Web administrators and causes the link Manage iBot Sessions to appear on the Analytics Administration page. Clicking this link opens the iBot Session Management screen.

When no iBot sessions are currently active, a message on the iBot Session Management screen alerts you.

When one or more iBot sessions are active, information about each iBot session appears, such as the Job ID and the Instance ID assigned to the iBot session by the Scheduler. Clicking the link in the Primary iBot column for an iBot session opens a popup window and navigates to the iBot session's definition in Siebel Delivers.

Expanding the iBot session shows the individual iBots (one iBot, or multiple iBots if they are chained). The state of the iBot is one of the following:

- Created
- Populated
- Conditional Request Resolved

Expanding a specific iBot in a particular session shows the recipients for the iBot and their type, such as the Engineering recipients defined in a group, or individual users. When the recipient is a group, the individual members of the group are not listed. Clicking the link in the Path column for an individual iBot opens a popup window and navigates to the iBot's definition in Siebel Delivers.

NOTE: When iBots are chained, the recipient list is dependent upon the parent iBot. The recipients are shown for the parent iBot definition only, and not for the actual execution of chained iBots.

To view information about active iBot sessions

1. In Siebel Analytics Web, do one of the following:
   - If you are running Siebel Analytics stand-alone, click the Admin link.
   - If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.
2. Click the following link:

   Manage iBot Sessions

   NOTE: If this link is not available, you do not have permission to view information about active iBot sessions.

   The iBot Session Management screen appears. A message alerts you if no iBot sessions are currently active.
3. To sort iBot sessions by their values in a particular column, click the sort button for that column.
4. To view more information about an iBot session, click the expand button.
5. To view more information about iBots within a particular session, click the expand button.
6. To view the definition of an iBot session or an individual iBot in Siebel Delivers, click the link for the session or iBot.

5 Administering Siebel Intelligence Dashboards

End users with appropriate privileges can modify personal and shared Siebel Intelligence Dashboards, including the addition of pages and content. End users cannot create dashboards.

Siebel Analytics Web administrators can create and manage dashboards, using the procedures in this chapter. For an introduction to dashboards and end-user procedures for modifying them, see *Siebel Analytics User Guide* and *Analytics Web Online Help*.

This chapter contains the following topics:

NOTE: For information about specifying view defaults for dashboards, see "Specifying View Defaults for Siebel Answers and Siebel Intelligence Dashboard Users" on page 43.

Creating and Deleting Shared Siebel Intelligence Dashboards

Before you create shared dashboards, make sure you have planned your Web Catalog directory or folder structure and security strategy. Guidelines for creating a shared dashboard, within the broader context of Web Catalog structure and security framework, are given in "Guidelines for Configuring Siebel Analytics Web Security for the Web Catalog and Dashboards" on page 116.

For more information about shared folder structures in the Web Catalog see Chapter 6, "Administering the Siebel Analytics Web Catalog."

For more information about permissions, see Chapter 7, "Managing Siebel Analytics Web Security."

Overall, to create a shared dashboard, you first create the dashboard and then add content using the Dashboard Editor. You can also assign Web Groups permissions to access the dashboard. Users who are members of more than one Web Group can select the dashboard they see by default from all of the dashboards to which they have permissions.

NOTE: When you use a Siebel operational application, dashboard integration is done using Siebel applications. For information, see *Siebel Analytics Installation and Configuration Guide*.

This section describes, from an administrator's perspective, how to create and delete dashboards, and add sections. For more information about adding pages, sections, and content from an end-user's perspective, see *Siebel Analytics User Guide*.

To create a shared dashboard

1 In Siebel Analytics Web, do one of the following:

- If you are running Siebel Analytics stand-alone, click the Admin link.
- If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.

2 Click the following link:

Manage Intelligence Dashboards

3 Scroll down if necessary, and click the following link:

Create a new Intelligence Dashboard

The Create Dashboard screen appears.

4 Specify entries for the following fields:

a Select the appropriate Group Folder from the drop-down list.

The members of this Web group (and other groups and users with appropriate permissions) will have Read access to the shared dashboard.

b Type a name for the dashboard in the Dashboard Name text box.

c In the Dashboard Builder text box, type the name of the user or Web Group that can make changes to the dashboard.

For information about creating Web Groups, see "About Siebel Analytics Web Groups and Siebel Analytics Session Variables" on page 104.

5 Click Finished when you are done.

6 In Siebel Answers, click the Dashboards tab in the selection pane, and then click the Refresh Display link near the bottom of the selection pane.

The newly created dashboard appears in the list of dashboards.

7 In Siebel Intelligence Dashboards, navigate to the dashboard and click the Edit Dashboard link.

The Dashboard Editor screen opens, where you can add content to the dashboard. For information, see *Analytics Web Online Help*.

To delete a dashboard

1 In Siebel Analytics Web, do one of the following:

- If you are running Siebel Analytics stand-alone, click the Admin link.
- If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.

2 Click the following link:

Manage Intelligence Dashboards

3 Locate the dashboard you want to delete and click the Delete button.

The Confirm Deletion screen appears.

4 Click Yes, and then click Finished.

Changing Siebel Intelligence Dashboards Properties

You can change dashboard properties at the Dashboard Properties screen.

To change dashboard properties

1 In Siebel Analytics Web, do one of the following:

- If you are running Siebel Analytics stand-alone, click the Admin link.
- If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.

2 Click the following link:

Manage Intelligence Dashboards

3 Locate the dashboard you want to change and click the Properties button.

The Dashboard Properties screen appears.

4 Follow directions in the table to change the property, and then click Finished when you are done.

NOTE: Clicking the Cancel button does not undo changes in the Dashboard Pages area.

Dashboard Property Change	Directions
Style	In the General Properties area, select the style from the Style drop-down list. For more information about styles, see Chapter 10, "Customizing the Siebel Analytics Web User Interface."
Hide Dashboard	Click this option to hide the dashboard from users. This is useful while you are setting up a dashboard.
Description	Type a description for the dashboard. The description does not appear on the dashboard.
Name of a page	In the Pages table, identify the page and click the Rename button.
Deletion of a page	In the Pages table, identify the page and click the Delete button.

Dashboard Property Change	Directions
Permissions for a page	In the Pages table, identify the page and click the Security button. For information about changing dashboard permissions, see "About Setting Siebel Analytics Web Permissions" on page 107.
Order in which the page tabs appear	In the Pages table, identify the page and click the Move Up or Move Down arrow to change the order in which the page appears.

Creating Siebel Intelligence Dashboards Action Links

Siebel Analytics action links allow end users to navigate from their analytic dashboards to a record in a Siebel operational application. Siebel Analytics applications contain prebuilt action links in requests and dashboards. For example, a user can drill down directly from a Siebel Sales Analytics dashboard to a specific record in a Siebel Sales view. Drilldown is based on the row identifier column contained in a request.

NOTE: End users must have the appropriate permissions and responsibilities to access the view, and any drilldown links.

To create an action link

1. Identify the target Siebel operational application view.
2. Select the Siebel operational application view or applet that you want to drill down to, and select Help > About View.

   A pop-up window shows the names of the view and applet. Note the view and list applet names. You need these names in Step 5.
3. Using a Subject Area that is appropriate for the target applet, use Siebel Answers to create a new Analytics request with the row identifier column in it, for example, Account_Row_ID.
4. Click the Properties button for the column.

   The Edit Column Format screen appears.
5. Make the following changes:
   a. Clear the option Default Data Format.
   b. Select the option Treat Text as Custom Text Format.
   c. Enter the following HTML, substituting the appropriate view name and list applet names. The following example HTML shown uses All Account List View as the View Name, and Account List Applet as the target applet:

      ```
      [HTML] "<a href=\"javascript:NQSWENav('All Account List View','Account List Applet','"@"'); \">"@" </a>"
      ```

6 Click Finished, and then view the request on the dashboard.

After the request is saved and placed on a dashboard, the row identifier column automatically produces an action link to the Siebel operational application.

Optionally, the row identifier column can be replaced with icons in the column, using the following sample HTML. This provides an image source for the action links, such as ActionLink.gif as shown in the sample. Substitute the name of the image that you want to use. The image must be located on the Web server machine:

```
[HTML] "<a href=\"javascript:NQSWENav('All Account List View','Account List
Applet','"@"'); \" title=\""@"\"><img src=\"Res/s_Siebel7/Views/ActionLink.gif\"
border=0></a>"
```

Setting the Number of Siebel Intelligence Dashboard Names to Appear on a Screen

Siebel Analytics typically displays the names of individual dashboards across the top of the screen. To minimize the amount of screen space used for this, when there are more than 15 dashboard names to display, Siebel Analytics creates a drop-down list from which users can choose the dashboard to view. A drop-down list shows individual dashboards as grouped under the name of the folder that contains them, and replaces the dashboard names at the top of the screen with that folder name. Together with the entry described in this section, you can create and manipulate folders containing dashboards within the Web Catalog to create the desired dashboard presentation.

You can change the number of dashboard names to show before a drop-down list appears by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. The minimum value is 1.

NOTE: This entry takes effect for all folders that contain dashboards if any folder contains more than the value you specify. For example, if dashboard folder A contains 8 dashboards and dashboard folder B contains 11 dashboards, and you specify a value of 10, drop-down lists will appear for both folders.

For information about working in the configuration file instanceconfig.xml, see “Making Siebel Analytics Web Configuration Changes” on page 11.

The following entry is an example:

```
<DashboardMaxBeforeMenu>10</DashboardMaxBeforeMenu>
```

Controlling Access to Saved Selection Options in Siebel Intelligence Dashboards

This section provides an overview of saved selections and information about administering saved selections. It contains the following topics:

Overview of Saved Selections in Dashboards

In Siebel Intelligence Dashboards, saved selections allow users to view dashboard pages with their most frequently used or favorite choices for filters and prompts, without the need to make choices manually for all of the prompts and filters that appear on the dashboard page.

Users and groups with the appropriate permissions and dashboard access rights can perform the following activities:

- Save various combinations of choices for filters and prompts as saved selections, for their personal use or use by others.
- Specify a saved selection as the default selection for a dashboard page, for their personal use or use by others.
- Switch between their saved selections.

You can restrict this behavior in the following ways:

- Users can view only the saved selections assigned to them.
- Users can save selections for personal use only.
- Users can save selections for personal use and for use by others.

NOTE: For information about end user use of saved selections, see the section about creating and assigning personal and shared saved selections for a dashboard page in *Siebel Analytics User Guide*.

Administering Saved Selections

This section describes the privileges and permissions required to administer saved selections. It also describes the relevant portions of the Web Catalog related to storing and administering saved selections.

Privileges for Saved Selections

In Siebel Analytics Web Administration, the following privileges in the Siebel Intelligence Dashboards area, together with permission settings for key dashboard elements, control whether users or groups can save or assign selections:

- Save Selections
- Assign Default Selections

You can set neither privilege, one privilege, or both privileges for a user or group, depending on the level of access desired. For example, a user who has neither privilege can view only the saved selection assigned as his or her default selection.

Permissions for Saved Selections

This section describes the permissions required for dashboard pages to administer saved selections, and the relevant portions of the Web Catalog structure for setting permissions on shared and personal saved selections.

Assigning Permissions to Dashboards

Permissions for dashboards, such as Read or Change/Delete, are set at the Manage Dashboards screen, available by clicking the link Manage Intelligence Dashboards at the Siebel Analytics Web Administration screen. The pages on the dashboard inherit the permissions set for a user or group.

Assigning Permissions for Saved Selections on a Dashboard Page

Permissions for assigning saved selections on a particular dashboard page are set at the Dashboard Properties screen, available by clicking the Dashboard Properties button in the Dashboard Editor.

When selection security is enabled at the Dashboard Properties screen by clicking the Enabled link, two buttons appear in the Selection Security column:

- The left button controls who can save shared selections for that dashboard page.
- The right button controls who can assign default selections for that dashboard page.

Clicking each button navigates to the appropriate location in the Web Catalog for that object. Web Catalog objects and permissions scenarios are described in more detail in the sections that follow.

Web Catalog Folder Structure for Saved Selections

In addition to the privileges set in Siebel Analytics Web Administration, the level of control that users and groups have over saved selections depends on their access rights to key elements. For example, users and groups that can create and edit underlying dashboards, save dashboard view preferences as selections, and assign selections to other users as default selections require Full Control permission to the key elements in shared storage, while users and groups that can view only their assigned default saved selections need only Read access to the key elements in shared storage.

Key elements in the Web Catalog include the following folders:

- Shared Storage Folders.

  Shared storage folders for dashboards are located within the _portal folder. Dashboards are identified by their assigned names.

  Permission settings control access to a specific dashboard for editing. Typically, if permissions are inherited down to the _selections and _defaults folders, users who can edit dashboards are also able to save selections and set defaults. Access to a specific dashboard folder controls whether a user or group can edit the dashboard.

  The _selections folder within a dashboard folder contains a page identifier folder for each dashboard page. Shared saved selections are located within this folder. Access to the page identifier folder controls whether a user or group can see, save or edit selections for that page.

  The _defaults folder within a _selections folder contains assigned default selections. Each group that has an assigned default appears here. Access to this folder controls whether a user or group can assign defaults.

- Personal Storage Folders.

  Within a user's personal folder, the _selections folder contains an individual user's saved selections. Like the shared _selections folder, a personal _selections folder contains a page identifier folder for each dashboard page. The page identifier folder contains personal saved selections and a _defaultlink file that specifies a user's preference for the personal defaulted selection.

  A personal saved selection default overrides an assigned shared selection default.

NOTE: If a dashboard page with saved selections is removed from the system, the saved selections are also removed from the Web Catalog. If the underlying dashboard structure changes such that a saved selection is no longer valid when a user accesses it, the default content appears on the dashboard and a message alerts the user.

Table of Permission and Privilege Settings for Creating Saved Selections

Table 7 describes typical user roles and specific permission settings that can be granted to users for creating saved selections. The folder names listed in the Permission and Privilege Settings column are described in the preceding section.

Table 7. User Roles and Permission Settings for Saved Selections

User Role	Permission and Privilege Settings
Power users such as IT users who need to perform the following tasks: ■ Create and edit underlying dashboards. ■ Save dashboard view preferences as selections. ■ Assign selections to other users as default selections.	In the Shared section of the Web Catalog, requires Full Control permission to the following folders: ■ dashboard_name. ■ _selections. ■ _defaults. Typically, no additional privileges need to be assigned.
Technical users such as managers who need to perform the following tasks: ■ Save selections as selections for personal use. ■ Save selections for use by others. Users cannot create or edit underlying dashboards, or assign view selections to others as default selections.	In the Shared section of the Web Catalog, requires Read permission to the following folders: ■ dashboard_name. In the Shared section of the Web Catalog, requires Write permission to the following folders: ■ _selections. ■ _defaults. Typically, no additional privileges need to be assigned.

Table 7. User Roles and Permission Settings for Saved Selections

User Role	Permission and Privilege Settings
Everyday users who need to save selections for personal use only.	In Siebel Analytics Web Administration, requires the following privilege to be set: ■ Save Personal Selections. In the dashboard page, requires that the following option is set: ■ Allow Saved Selections. In the Web Catalog, no additional permission settings are typically required.
Casual users who need to view only their assigned default selection.	In the Shared section of the Web Catalog, the user needs Read permission to the following folders: ■ dashboard_name. ■ _selections. ■ _defaults. In the Web Catalog, no additional permission settings are typically required.

Example Usage Scenario for Saved Selection Administration

Depending on privileges set and permissions granted, you can achieve various combinations of user and group rights for creating, assigning, and using saved selections.

For example, suppose a group of power users cannot change dashboards in a production environment but are allowed to create saved selections and assign them to other users as default selections. The following permission settings for the group are required:

- Read access to the dashboard, using the Manage Intelligence Dashboards administration screen.
- Change/Delete access to the _selections and _defaults subfolders within the dashboard folder in the Web Catalog, assigned using the Dashboard Properties screen accessible from the Dashboard Editor.

Setting the Number of Siebel Analytics Briefing Book Links to Follow

A Briefing Book navigation link is a special type of link that can be added to a dashboard using the Dashboard Editor. The default value for the maximum number of links to follow is 5.

You can change the default by modifying the Siebel Analytics Web configuration file instanceconfig.xml to add the following entry. The minimum value is 1. The maximum value is 10.

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

The following entry is an example:

```
<MaxFollowLinks>10</MaxFollowLinks>
```

Downloading Siebel Analytics Results in Non-UNICODE Format

Siebel Analytics provides options for downloading results that appear in Siebel Answers as options for the Download link. The Download link can also appear with a request in a dashboard.

By default, the Download Data option downloads results in a UNICODE, tab-separated text file. For organizations that require a non-UNICODE, comma-separated file for use with certain applications, you can override behavior of the Download Data option or add another download option by modifying the message kmsgEVCDownloadLinks in the XML messages file viewscontrolmessages.xml.

If you change the behavior of the Download Data option or add a new option that retrieves comma-separated data, the instanceconfig.xml entry described in this section is used to determine the character set to use. You can view supported character sets by examining the message file charactersetdefinitions.xml located in the SiebelAnalyticsData\Web\Config directory.

For example, adding the following XML code to the message kmsgEVCDownloadLinks in the file viewscontrolmessages.xml file adds the option Download CSV to the Download link:

```
          <a class="NQWMenuItem" name="SectionElements"
href="javascript:void(null);" onclick="NQWClearActiveMenu();
Download('@{command}&amp;Format=csv&amp;Extension=.csv')">Download CSV</a>
```

The Download CSV option will download a comma-separated file that uses the following entry to determine the character set.

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

The following entry is an example:

```
<CSVCharset>us-ascii</CSVCharset>
```

For information about making changes to the XML messages files, see "Customizing the Siebel Analytics Web User Interface Using XML Message Files" on page 196.

Integrating Siebel Answers into Other Portals or Intranets

The following customization settings are available for integrating Siebel Answers into other portals and intranets without requiring the use of Siebel Intelligence Dashboards. These settings are configured in XML message files.

You can perform the following actions:

- Change the text of the Dashboards link.
- Change the URL to which users are directed when they click that link.

NOTE: Make sure that you first review the information in "Customizing the Siebel Analytics Web User Interface Using XML Message Files" on page 196 before making any integration changes.

To change the text of the dashboards link

1. Navigate to the file UIMessages.xml.

   This file is located in the folder Web\App\Res\l_xx\Messages in the Siebel Analytics Web installation directory, where xx is the language identifier of the selected locale.

   CAUTION: Always make a backup copy of the file UIMessages.xml before you make any changes.

2. Use a text editor to open the file UIMessages.xml.
3. Locate the message "kmsgUIPortal."

   The message has the following form:

   ```
   <WebMessage name="kmsgUIPortal">
     <TEXT>Dashboards</TEXT>
   </WebMessage>
   ```

   The default text is Dashboards.

4. Change the default text to text of your choosing.

   For example, to change the text to be your company's intranet, you might change it to the following:

   ```
   <WebMessage name="kmsgUIPortal">
     <TEXT>Intranet</TEXT>
   </WebMessage>
   ```

5. Save the file when you are done.

   Your changes take effect when the Analytics Web Server service is restarted.

To change the destination of the dashboards link

1 Navigate to the file ControlMessages.xml.

This file is located in the folder Web\App\Res\l_xx\Messages in the Siebel Analytics Web installation directory, where xx is the language identifier of the selected locale.

CAUTION: Always make a backup copy of the file ControlMessages.xml before you make any changes.

2 Use a text editor to open the file ControlMessages.xml.

3 Locate the message "kmsgPortalLink."

The message has the following form:

```
<WebMessage name="kmsgPortalLink">
 <HTML>
   <A insert="1">
    <MessageRef name="kmsgUIPortal" />
   </A>
 </HTML>
</WebMessage>
```

The default location is Siebel Intelligence Dashboards (indicated by the reference to insert="1", which is an internal reference).

4 Change the default location to a location of your choosing.

NOTE: If no location is specified, no link appears in Siebel Answers.

For example, to change this to your company's intranet, you can include the appropriate attributes and change it to point to that location instead:

```
<WebMessage name="kmsgPortalLink">
 <HTML>
   <A href="http://intranet" target="_top" title="Click here for your intranet">
    <MessageRef name="kmsgUIPortal" />
   </A>
 </HTML>
</WebMessage>
```

5 Save the file when you are done.

Your changes take effect when the Analytics Web Server service is restarted.

For more information about controlling the default appearance and behavior of the Siebel Analytics Web user interface, see "Customizing the Siebel Analytics Web User Interface Using XML Message Files" on page 196.

6 Administering the Siebel Analytics Web Catalog

This chapter describes how to administer the Siebel Analytics Web Catalog and provides information about basic maintenance procedures. For organizations that have Siebel Analytics applications, this chapter also explains how to use Analytics Catalog Manager to locate the names and descriptions of the requests delivered with preconfigured dashboards.

This chapter contains the following topics:

Changing the Name and Location of the Siebel Analytics Web Catalog

The installation process sets the name of the Web Catalog to default.webcat for Siebel Analytics stand-alone sites, and to SiebelAnalytics.webcat for organizations that have Siebel Analytics applications.

The following locations are the default locations for the Web Catalog:

- In Windows

  `\SiebelAnalyticsData\Web\Catalog`

- In UNIX

  `INSTALLDIR/Data/web/catalog`

  where INSTALLDIR is the Siebel Analytics Web installation directory. The default installation directory is /usr/local/SiebelAnalytics.

If you move the Web Catalog or change its name, you need to update the Siebel Analytics Web configuration file instanceconfig.xml to specify the new location or name.

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

The following entry is an example:

```
<CatalogPath>/usr/local/SiebelAnalyticsData/web/catalog/test.webcat</CatalogPath>
```

NOTE: There is a one-to-one relationship between a Siebel Analytics repository and a Siebel Analytics Web Catalog. Multiple Siebel Analytics Web servers may be configured to point to a single Siebel Analytics server, but each Web Catalog must point to a unique repository. You cannot, for example, configure two Siebel Analytics Web servers to use the same Web Catalog concurrently.

How the Siebel Analytics Web Catalog Backup Process Works

Siebel Analytics Web maintains automatic Web Catalog backup files and version files in the location where the Web Catalog is stored. Web Catalog backup files are useful when the Siebel Analytics Web administrator is customizing or testing Siebel Analytics Web content. The version file is used to help maintain Web Catalog integrity.

This section contains the following topics:

About Siebel Analytics Web Catalog Backup Files

By default, Siebel Analytics Web performs a Web Catalog backup every five minutes. The backup files are saved into the same location as the in-use Web Catalog. Automatic backups are identified by nnn.autosave appended to the Web Catalog name, where nnn represents the sequential number assigned to the backup.

For example, in Windows, if the path to the Web Catalog is:

```
c:\SiebelAnalyticsData\Web\Catalog\SiebelAnalytics.webcat
```

then the third backup file is:

```
c:\SiebelAnalyticsData\Web\Catalog\SiebelAnalytics.webcat.03.autosave
```

The default number of Web Catalog backup files is 10. If the server crashes, it creates another set of 10 backup files. After ten sets of backup files have been created, the numbering goes back to 01-10.

You can control the number of automatic Web Catalog backup files and the interval between them. For more information, see "Setting the Siebel Analytics Web Catalog Backup Parameters" on page 72

About the Siebel Analytics Web Catalog Version File

Information about the last known good Web Catalog backup is stored in a file ending in `.version`. For example, in Windows, using the example locations shown in "About Siebel Analytics Web Catalog Backup Files" on page 70, the current version of SiebelAnalytics.webcat would be tracked in:

```
c:\SiebelAnalyticsData\Web\Catalog\SiebelAnalytics.webcat.version
```

The version file is interrogated during Siebel Analytics Web startup:

- If the version file does not exist, the Web Catalog is read from the original path specified in the CatalogPath registry key. (For more information about this key, see "Changing the Name and Location of the Siebel Analytics Web Catalog" on page 69.)
- If the path does not exist, the Web Catalog is created in the default location for the operating system, described in "Changing the Name and Location of the Siebel Analytics Web Catalog" on page 69.
- If the version file exists, the backup file that it references (nnn.autosave) must also exist. If the backup file does not exist, Siebel Analytics Web will not start.

The version file is updated during Siebel Analytics Web shutdown. After a successful shutdown, the version file includes an entry indicating that the Web Catalog was successfully written to the original catalog path.

NOTE: The Siebel Analytics Web shutdown process does not create a Web Catalog backup file.

When Changes Are Written to the Siebel Analytics Web Catalog

Changes made to the Web Catalog are written to the Web Catalog when the Analytics Web Server service is stopped. Until the service is stopped, changes are written to the current Web Catalog backup file.

When the Analytics Web Server service is started again, the Web Catalog is read, and any subsequent changes are written to the current backup file.

Setting the Siebel Analytics Web Catalog Backup Parameters

Siebel Analytics Web uses a combination of three parameters to determine how many backup files to maintain and which ones to replace with the newest information. If the oldest file does not exceed the CatalogAutoSaveTimeSpanDays parameter, it is not always the one replaced. Siebel Analytics Web uses other factors to determine which one is least useful and replaces that one. Table 8 lists and describes these parameters.

Table 8. Siebel Analytics Web Catalog Backup Parameters

Parameter	Description
CatalogAutoSaveMinutes	Time interval (in minutes) between backups.
CatalogAutoSaveTimeSpanDays	Siebel Analytics Web checks the oldest backup file against this parameter and if the file's timestamp exceeds this time span, then the file is replaced. If no files exceed this time span, then Siebel Analytics Web determines the least useful file and replaces it. This determination uses file age as a weighted factor.
CatalogMaxAutoSaves	The maximum number of backup files to retain.

To edit these parameters, you can work in the instanceconfig.xml file. The edits follow this form:

```
<CatalogMaxAutoSaves>20</CatalogMaxAutoSaves>
```

For information about working in the configuration file instanceconfig.xml, see "Making Siebel Analytics Web Configuration Changes" on page 11.

Creating a New Siebel Analytics Web Catalog

The following procedure explains how to create a new Siebel Analytics Web Catalog.

To create a new Web Catalog

1 Stop the Analytics Web Server service.

NOTE: When Analytics Web is installed on a Microsoft IIS Web server, the IIS Web server creates temporary files for its Web caching capability in a location specified by the Microsoft IIS installation. If the Analytics Server is stopped before the Analytics Web Server, these temporary files usually remain, consuming disk space.

2 Stop these other services:

- Siebel Analytics Server

- Siebel Analytics Scheduler
- Siebel Analytics Cluster Server
- IIS Admin Service (Microsoft Internet Information Server), and any additional services that it wants to stop

3 Set the CatalogPath registry key or instanceconfig.xml entry to point to a new Web Catalog path that does not already exist.

For details, see "Changing the Name and Location of the Siebel Analytics Web Catalog" on page 69.

4 Make sure that there is no version file associated with the new Web Catalog and path and that the new target directory is empty.

5 Restart the services.

Recovering from a Failure to Load the Siebel Analytics Web Catalog

Siebel Analytics Web attempts to load the Web Catalog as part of its startup procedures. If it cannot load the Web Catalog, a failure message is logged in the operating system's application event log, and no one will be able to access Siebel Analytics Web. When possible, the system attempts to load the last saved Web Catalog, determined from the version file. (For information about the version file, see "How the Siebel Analytics Web Catalog Backup Process Works" on page 70.)

To recover from a failure to load the Web Catalog

1 Stop the Analytics Web Server service.

NOTE: When Analytics Web is installed on a Microsoft IIS Web server, the IIS Web server creates temporary files for its Web caching capability in a location specified by the Microsoft IIS installation. If the Analytics Server is stopped before the Analytics Web Server, these temporary files usually remain, consuming disk space.

2 Stop these other services:

- Siebel Analytics Server
- Siebel Analytics Scheduler
- Siebel Analytics Cluster Server
- IIS Admin Service (Microsoft Internet Information Server), and any additional services that it wants to stop

3 Make a backup of files related to the Web Catalog.

4 Rename the latest backup file (nnn.autosave) to the name of the original Web Catalog.

For example, rename SiebelAnalytics.webcat.03.autosave to SiebelAnalytics.webcat.

5 Delete the associated version file (.version) file.

6 Restart the services.

If the Web Catalog still fails to load, repeat Step 3 through Step 6 with earlier backup files until Siebel Analytics Web starts successfully.

Moving a Siebel Analytics Web Catalog to Another System

This procedure explains how to move an existing Web Catalog to another Analytics installation. (For information about migrating a complete Siebel Analytics installation, see the Technical Notes section on SupportWeb.)

To move an existing Web Catalog to another system

1 Stop the Analytics Web Server service.

2 Stop these other services:

- Siebel Analytics Server
- Siebel Analytics Scheduler
- Siebel Analytics Cluster Server

NOTE: When Analytics Web is installed on a Microsoft IIS Web server, the IIS Web server creates temporary files for its Web caching capability in a location specified by the Microsoft IIS installation. If Siebel Analytics Server is stopped before Siebel Analytics Web, these temporary files usually remain, consuming disk space.

When you stop the Analytics Web Server service, the Web Catalog is written to the path shown in the CatalogPath registry key or instanceconfig.xml entry. (For more information, see "Changing the Name and Location of the Siebel Analytics Web Catalog" on page 69.)

3 Verify successful shutdown by checking that the latest modified Web Catalog file is the original Web Catalog file with a .webcat extension, and not a backup file with an nnn.autosave extension.

4 Copy the original Web Catalog file to the appropriate location on the new system.

Optionally, you can also copy the version file (it has a .version extension).

NOTE: If you are migrating a Siebel Analytics environment to a new system, make sure you also migrate the Siebel Analytics Server repository file and the Scheduler tables. The Scheduler tables are required for iBots.

5 Restart the services.

NOTE: If you do not want to stop the services, you can copy the related Web Catalog files to the appropriate location on the new system. This includes the original Web Catalog file (.webcat extension), the Version file (.version extension), and the Web Catalog backup files (nnn.autosave extensions).

Migrating Changes to Siebel Analytics Requests and Filters

When you create and save requests and filters in Siebel Analytics Web, the objects are saved in the Web Catalog. The requests and filters are saved with the logical object references as they exist in the Siebel Analytics metadata in the Presentation layer of the Siebel Analytics Server Administration Tool.

If the Presentation layer metadata that affects these is changed through the renaming of objects, aliases are maintained for these objects that the Web Catalog can reference.

However, if the Presentation layer metadata changes through the following ways, the link from the stored object in the Web Catalog to the metadata will be broken:

- Aliases are deleted.
- Objects are moved between Presentation Catalog Folders in the Presentation layer.

To repair those links within a stored request, you need to refresh the requests and filters.

To refresh requests and filters

1. Log on to Siebel Analytics Web and access Siebel Answers.
2. Open the broken request and click the Advanced tab.
3. Click the Enter SQL link.
4. Change the FROM clause to reference the new Subject Area (Catalog Folder in the Presentation layer) where the SELECT columns now reside.
5. If the column names have been changed, those changes must also be applied.
6. Reapply any required filters using the new Presentation layer definitions.
7. Redefine any affected saved filters using the new Presentation layer definitions.

   The changes are applied the next time that the Analytics Web Server service and the IIS Admin Service are started.

About Siebel Analytics Catalog Manager

Siebel Analytics Catalog Manager Siebel is a tool for Siebel Analytics Web administrators to perform offline management of the Siebel Analytics Web Catalog. It should be installed on a secure machine accessible only to Siebel Analytics Web administrators. Administrators can use Analytics Catalog Manager to perform the following functions:

- Administrators can rename and delete content, and move and copy content within and between Web Catalogs. This is critical for installations that maintain distinct test and production systems, as well as those that are creating content on top of Siebel Analytics Stand-Alone.
- Administrators that have one or more Siebel Analytics applications installed can use Analytics Catalog Manager to generate a list of dashboard requests. The requests are distributed with Siebel Analytics applications.

- Analytics Catalog Manager allows administrators to upgrade their Siebel Analytics applications to new versions without losing any site-specific changes and enhancements.
- Administrators that need to localize content can use Analytics Catalog Manager to export text strings for translation.

Opening Siebel Analytics Catalog Manager

Use only the procedure that follows to open Analytics Catalog Manager. Analytics Catalog Manager operates on the Web Catalog only when the Analytics Web Server service is stopped.

To open Analytics Catalog Manager

1. Stop the Analytics Web Server service.

   NOTE: When Analytics Web is installed on a Microsoft IIS Web server, the IIS Web server creates temporary files for its Web caching capability in a location specified by the Microsoft IIS installation. If the Analytics Server is stopped before the Analytics Web Server service, these temporary files usually remain, consuming disk space.

2. Stop these other services:
   - Siebel Analytics Server
   - Siebel Analytics Scheduler
   - Siebel Analytics Cluster Server
   - IIS Admin Service (Microsoft Internet Information Server), and any additional services that it wants to stop.
3. On the machine where Analytics Catalog Manager is installed, choose Start >Programs > Siebel Analytics > Siebel Analytics Catalog Manager.

   The Analytics Catalog Manager window opens.
4. When you have finished working with Analytics Catalog Manager, restart the services.

Viewing the Siebel Analytics Catalog Manager Workspace

This section explains how to how to view items in the Analytics Catalog Manager workspace. It contains the following topics:

How the Siebel Analytics Catalog Manager Workspace Appears

Figure 1 shows how the Analytics Catalog Manager workspace appears after opening SiebelAnalytics.webcat and clicking the Web Catalog icon. (SiebelAnalytics.webcat is the Web Catalog distributed with Siebel Analytics applications.)

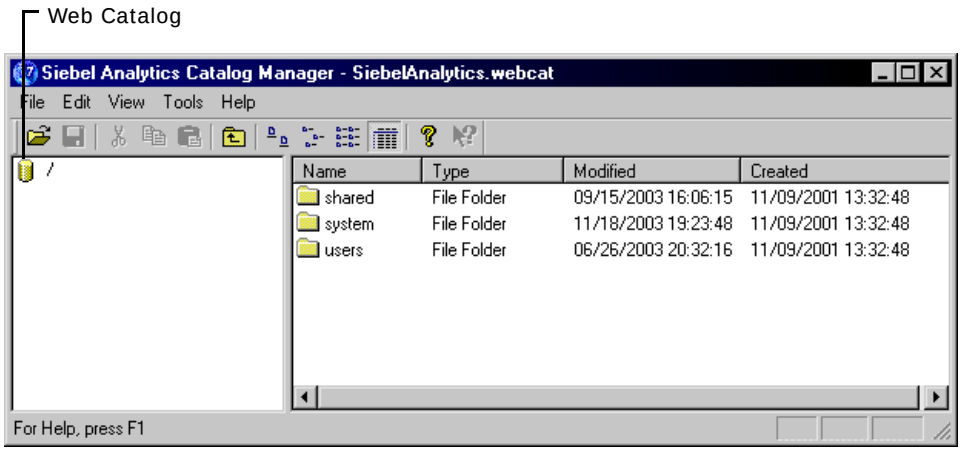


Figure 1. Analytics Catalog Manager Workspace

To view the Web Catalog tree structure

- Perform one of the following actions:
  - Double-click the Web Catalog icon.
  - Click a folder in the right pane.

About Folders in the Siebel Analytics Catalog Manager Workspace

The shared folder contains content that is shared among Analytics Web users. This includes the preconfigured dashboards and requests that are distributed with Siebel Analytics applications, and other items such as shared filters.

The system folder contains administrative elements of Siebel Analytics Web. Some of these elements are distributed with the product, and others are set up by the Siebel Analytics Web administrator, such as privileges.

The users folder contains content that Analytics Web users with the appropriate permissions have saved to their personal folders, such as individual requests.

About Siebel Analytics Catalog Manager Columns in the Workspace

The right pane of the Analytics Catalog Manager has four columns. One column, needing additional explanation, is described here.

Type. Identifies the type of item. Items that are identified as “unknown file” are generally internally-used items, and their type is not exposed in Analytics Catalog Manager.

Working with Siebel Analytics Catalog Manager Properties

Analytics Catalog Manager allows Siebel Analytics Web administrators to work with a subset of the administration functions available in Siebel Analytics Web. For example, you can modify an item's properties, hide a request from display, and change user and group permissions to items in the Web Catalog.

For more information about setting permissions, see "Setting Siebel Analytics Web Permissions Through Analytics Catalog Manager" on page 112.

Other administration tasks, such as setting up new Web Groups, are performed using the administration features in Siebel Analytics Web.

To view properties for an item

- Double-click the item, or right-click the item in the Name list and choose Properties.

  The Properties window for the item appears.

  NOTE: The New button is used to create a new property. You should use it only if instructed to do so by Siebel Systems. The Read Only option is unused. The System option indicates that the item is maintained internally and should not be altered.

Figure 2 is an example of the Properties window for a request named Top Partner Opportunities. This request is from the Overview page on the Channel Sales dashboard in the Siebel Partner Analytics application.

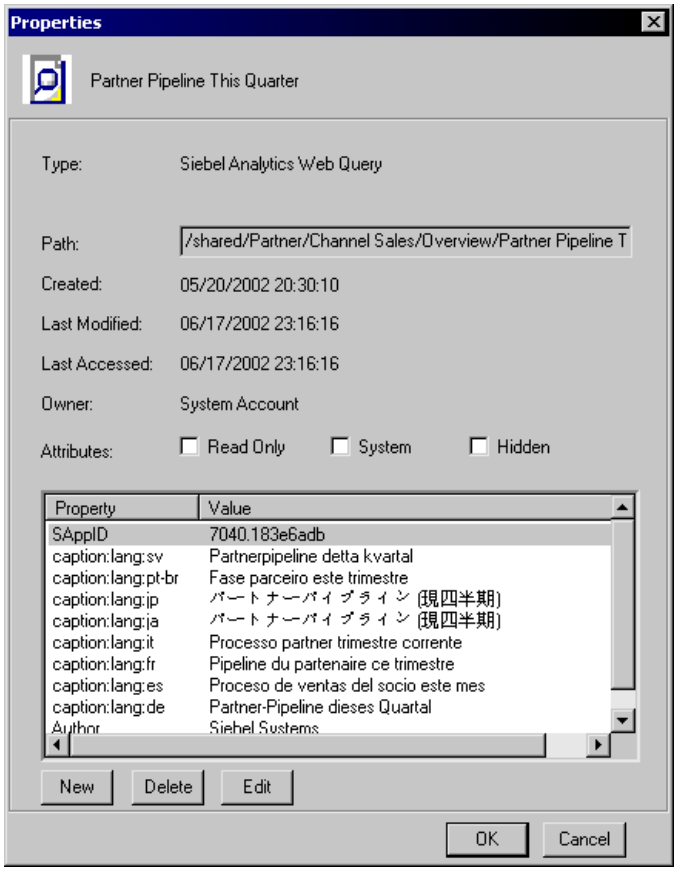


Figure 2. Properties Window

To hide an item from display in Siebel Analytics Web

- Select the Hidden option.

Copying and Pasting Content Between Siebel Analytics Web Catalogs

This section describes how to move content between Web Catalogs.

Web Catalogs are structured in hierarchical folders. When copying or merging items, remember to also copy any items that are associated with them, such as dashboard folders, request links, and requests. URL paths may need to be re-established after a copy or merge operation if the entire folder path is not copied, for example, if added to the dashboard as a link or text.

NOTE: If you want to upgrade your existing Web Catalog with a newer version received from Siebel Systems, see "Upgrading the Siebel Analytics Web Catalog to a Newer Version" on page 86.

To copy and paste content between Web Catalogs

1. Make backup copies of both Web Catalogs.
2. Choose File > Open from the Analytics Catalog Manager toolbar and select the first Web Catalog to open.

   This should be the Web Catalog that you want to change. The name of the Web Catalog that you opened appears in the title bar.

   TIP: It is recommended that you first open the Web Catalog that you want to change. If the catalogs have the same name, rename the second catalog before opening it.
3. Using the same instance of Analytics Catalog Manager, choose Choose File > Open from the Analytics Catalog Manager toolbar, and then select the second Web Catalog to open.

   This should be the Web Catalog that contains the content you want to copy.

   A second instance of Analytics Catalog Manager opens. The name of the second Web Catalog appears in the title bar of the second instance.
4. If necessary, reposition both instances of Analytics Catalog Manager on your screen so you can clearly see the title bars of both Analytics Catalog Manager instances.
5. To view the items that you can work with, expand the tree structure in the left pane, or double-click an item in the right pane.
6. To copy content, highlight the source item in the source (second) Web Catalog and choose Edit > Copy. Then, highlight the corresponding target item in the target (first) Web Catalog and choose Edit > Paste.

   You can also use drag-and-drop to copy and paste content.

Replicating Siebel Analytics Web Catalogs

Siebel Analytics can copy and merge contents of selected catalog folders between Siebel Analytics Web servers. The replication configuration can be split into individual tasks. The replication task is an instruction to periodically merge catalog contents for specified catalog folders from one server to another. Two-way replications are possible.

A typical scenario where replication is useful is when you use one Siebel Analytics Web server to prepare and publish shared reports, and use two or more production servers to support the users. In this scenario you share the catalog from Server 1 to two production servers and then share the catalogs on each productions server with the other. To accomplish this you set up the following replication tasks in the Siebel Analytics Web Replication Agent's (sawrepa) configuration file:

- /shared: SAW1 to SAW2
- /shared: SAW1 to SAW3
- /users: SAW2 to SAW3
- /users: SAW3 to SAW2

When a Siebel Analytics Web server participates in replication, either as a publisher or subscriber, it tracks changes made to catalog items that are marked as "Replicate" and keeps them internally in the change log. Another Siebel Analytics Web server can make a SOAP call to request to export those changes to a file or to import and replay modifications recorded in a file exported earlier from another server. The Siebel Analytics Web Replication Agent uses SOAP to manage replication related activities on all Siebel Analytics Web servers and performs import and export operations. For information on the SOAP calls used, see "ReplicationService" on page 167.

Setup for Siebel Analytics Web Catalog Replication

To enable Web Catalog replication, set up the conditions described in this section.

Set UseReplication Property

By default, the replication functionality is disabled for Siebel Analytics Web servers. For a Siebel Analytics Web server to participate in replication, as either a publisher or subscriber, its UseReplication property must be set to "Y." You can set this either in the server's `instanceconfig.xml` file or in the registry.

Use Siebel Analytics Web Replication Agent

You must use the Siebel Analytics Web Replication Agent (sawrepa) on one server. This agent handles all the replication tasks. It has some command line options. For more information, see "Using the Siebel Analytics Web Replication Agent" on page 83.

Edit Siebel Analytics Web Replication Agent Configuration File

You need to edit the Siebel Analytics Web Replication Agent's configuration file (config.xml) to specify which folders on which Siebel Analytics Web servers are to be replicated. For more information, see "Editing the config.xml File" on page 84.

Copy Web Catalog Files

The Siebel Analytics Web Replication Agent copies changes made to a Web Catalog. Because it only copies changes, and not the entire contents, you must first make a copy of the source Web Catalog on each of the subscribing Siebel Analytics Web servers.

Adjust Replication Flags

Because the Web Catalog file came from another server its Replication flag maybe set incorrectly. Adjust it using the sawrepa mark command. Start by removing the Replication flag from all catalog nodes with this command:

```
sawrepa mark <newservername> /n /shared /users
```

Then set the Replication flag again with this command:

```
sawrepa mark <newservername>
```

Resuming Replication After Siebel Analytics Web Server Downtime

If a Siebel Analytics Web server is offline for a time period in excess of that configured with the LogExpiresHours entry in the configuration file, then it is not possible to synchronize its contents using the sawrepa run command. The Siebel Analytics Web Replication Agent then disables all corresponding replication tasks.

You can resume replication by copying over the Web Catalog folders that the server subscribes to, and then removing and restoring the Replication flags. However, this method loses the catalog contents that were not replicated to other servers. The following method preserves all catalog contents.

To resume replication

1. Use the CatalogBrowser utility to erase all replication logs (/system/replication node).
2. Start the server.
3. Remove the Replication flag using the sawrepa mark <servername> /n command.
4. Delete or rename folders that are replicated from other servers and copy them over from other servers using the sawrepa remotecopy command.
5. Restore the Replication flag using the sawrepa mark <servername> command.
6. Reenable replication tasks by editing the configuration file to delete the lastPerformed attribute from ReplicationTask elements that have the Siebel Analytics Web server as an import or export target.

Using the Siebel Analytics Web Replication Agent

The Siebel Analytics Web Replication Agent (sawrepa) is a command line utility that carries out the common replication tasks such as copy, export, import, mark for replication, and purge log.

The Siebel Analytics Web Replication Agent needs information about the Siebel Analytics Web servers and the replication tasks. This information is stored in the `config.xml` configuration file, which is located in the same directory as sawrepa executable.

The command line for the Siebel Analytics Web Replication Agent uses the following format:

```
sawrepa [/C path] command [command parameters]
```

The path following the /C specifies the path to the config.xml file. The command can be one of the following:

- mark
- purge
- remotecopy
- run

mark

Adds or removes the Replicate flag on specified folders on all or specified Siebel Analytics Web servers. Setting the Replicate flag does not mean that the corresponding catalog item is immediately replicated. Only future modifications are noted in the catalog log and subsequently get replicated. The syntax for the mark command is:

```
sawrepa [/C configfile] mark {all|servername} [/n] [catalogfolders]
```

- **all | servername.** The name of the Siebel Analytics Web server (or all) on which to run the mark command. If you use a specific server, the name you specify must match the name attribute of the corresponding server tag in the config.xml file.
- **/n**. If present indicates that the Replicate flag should be removed, otherwise it is added.
- **catalogfolders.** A list of folders on which to add or remove the Replicate flag. Separate each folder in the list with a space. If you do not specify any folders, then sawrepa executes the mark command on every folder for the specified servers listed in the config.xml file.

purge

Cleans up obsolete log entries on specified folders on all or specified Siebel Analytics Web servers. The configuration file contains an entry (LogExpiresHours) which defines the minimum time period for which a replication log should be preserved. The timestamp for determining obsolete items is set using this period. The syntax for the purge command is:

```
sawrepa [/C configfile] purge {all|servername} [catalogfolders]
```

- **all | servername.** The name of the Siebel Analytics Web server (or all) on which to run the purge command. If you use a specific server, the name you specify must match the name attribute of the corresponding server tag in the config.xml file.

- **catalogfolders.** A list of folders to clean. Separate each folder in the list with a space. If you do not specify any folders, then sawrepa executes the purge command on every folder for the specified servers listed in the config.xml file.

remotecopy

Exports the contents of the specified catalog folders from the source server and imports them in the destination server. The syntax for the remotecopy command is:

```
sawrepa [/C configfile] remotecopy sourceServer destinationServer catalogfolders
```

- **sourceServer.** The name of the source server as specified in the config.xml file.
- **destinationServer.** The name of the destination server as specified in the config.xml file.
- **catalogfolders.** A list of folders to clean. Separate each folder in the list with a space. If you do not specify any folders, then sawrepa executes the purge command on every folder for the specified servers listed in the config.xml file.

run

Executes all replication tasks that are setup in config.xml file and that have not expired. This command does not have any runtime arguments. The syntax for the run command is:

```
sawrepa [/C configfile] run
```

Replication tasks can expire if the replication log was cleaned after its last successful run. If that is the case manual intervention is required. An administrator can synchronize the contents of replicated folders using the remotecopy command, then purge the replication log and then reenable the task by deleting lastPerformed timestamp in the config.xml file.

Editing the config.xml File

The structure of the config.xml file is as follows:

```
<Config>
  <General>
    < ExportDirectory />
    < LogExpiresHours/>
  </General>
  <Server>
  <Folderset>
    <Folder>
  </Folderset>
  <ReplicationTask/>
</Config>
```

Table 9 describes the elements in the config.xml file.

Table 9. Elements in the config.xml File

Element	Parent	Occurrences	Description
Config	N/A	1	XML root element.
General	Config	1	Contains general settings applicable to all servers and replication tasks.
Export Directory	General	1	Contains the UNC path to the shared directory where export files should be placed. It should be accessible by the same name from all Siebel Analytics Web servers.
LogExpiresHours	General	1	Specifies the time in hours (as a double value) that the replication log should be preserved on all Siebel Analytics Web servers.
Server	Config	1…n	Defines connection information for each server. Its attributes are: ■ **name.** The logical id of the server, which is used to identify it in replication tasks and in the sawrepa command line. ■ **url.** The URL of the server, such as: http://localhost/analytics/saw.dll ■ **user.** The user name. ■ **pwd.** The password.
Folderset	Config	0…n	Defines a list of catalog folders. Its only attribute is: ■ **name**

Table 9. Elements in the config.xml File

Element	Parent	Occurrences	Description
Folder	Folderset	0…n	Adds a folder to the folderset. It specifies the full path to the catalog folder.
ReplicationTask	Config	0…n	Defines the replication tasks. Its attributes are: ■ **source.** The source server name. This must match a server defined in Server element. ■ **destination.** The destination server name. This must match a server defined in Server element. ■ **folders.** The name of the folders set. This must match a folder set defined in the Folderset element. ■ **lastPerformed.** The timestamp of the last successful run of this task. The sawrepa utility updates this value.

Upgrading the Siebel Analytics Web Catalog to a Newer Version

This section is for organizations that have Siebel Analytics operational applications and have customized Web Catalog content. It explains how to merge your current Web Catalog with a newer version received from Siebel Systems.

NOTE: If you made no changes to the Web Catalog distributed with previous versions, this section may not apply to you. You can begin using the SiebelAnalytics.webcat Web Catalog distributed with this version.

The update is done using Analytics Catalog Manager. The process compares the base Web Catalog distributed with the previous version with your customized Web Catalog to get a baseline of the changes, and then merges your existing Web Catalog content with the content from the previous version. This process preserves any site-specific changes that have been made to your Web Catalog, and allows Siebel Analytics Web administrators to accept or reject specific changes.

The update makes use of three Web Catalogs:

- The *Original Siebel Catalog*, which is the Web Catalog that you received with your currently-installed Siebel Analytics application. It is also distributed at the root level of the installation CD-ROM as the file SiebelAnalytics-n.n-GA.webcat, where n.n is the version number.
- The *Current Release Siebel Catalog*, which is the Web Catalog that is installed as the file SiebelAnalytics.webcat.
- The *Modified Customer Catalog*, which is the Web Catalog that you are currently using. In the following procedure, you will copy and rename this to SiebelAnalytics.webcat.old.

To update the Web Catalog to a newer version

1. Make a backup copy of your Modified Customer Catalog, rename it to SiebelAnalytics.webcat.old, and move it to a temporary location.

   This is to make sure that a copy of your current Web Catalog is available for use during the upgrade process.

2. Install the new software version received from Siebel Systems according to the instructions given in *Siebel Analytics Installation and Configuration Guide*.

   This installs the Current Release Siebel Catalog, named SiebelAnalytics.webcat.

   (If the software is already installed, you are prompted to uninstall it first. During the uninstall, if you select the option to keep your current Web Catalog, it is copied and renamed to SiebelAnalytics.webcat.old. You can use this Web Catalog, or the copy in the temporary location, during the upgrade process.)

3. Make sure that your current Siebel Analytics repository has been migrated to the current version.

   For instructions, see *Siebel Analytics Server Administration Guide*.

4. Copy the Original Siebel Catalog from the installation CD-ROM into the location that holds your Modified Customer Catalog.

5. Open Analytics Catalog Manager and open the Current Release Siebel Catalog, SiebelAnalytics.webcat.

6. Choose Tools > Upgrade Catalog.

   The Upgrade Current Catalog window opens.

7. In the Original Siebel Catalog area, browse to locate the Original Siebel Catalog, SiebelAnalytics-n.n-GA.webcat.

8. In the Modified Customer Catalog area, browse to locate your Modified Customer Catalog, SiebelAnalytics.webcat.old.

9. Click OK.

   Analytics Catalog Manager generates a baseline of the changes and compares your existing Web Catalog content with the content in the new version, merges in any changes, produces a request highlighting differences, and allows you to indicate how you want differences handled. If the Web Catalogs have conflicting content, you can choose which Web Catalog the content should be taken from.

   The end result is a merged Web Catalog that contains the site-specific changes, as well as new Siebel metadata. When the merge process completes, the Current Release Siebel Catalog that is loaded into Analytics Catalog Manager contains the merged content.

   The merge process may take several minutes. A message alerts you when the merge is complete.

   NOTE: The log file SiebelAnalyticsMigrationLog.txt holds information about the merge process. This log file is written to the location that holds the Web Catalog. If you see an error logged in the file, this means that the path in question had a problem that did not allow the merge mechanism to resolve the merge. No action was taken. If you want to merge that particular item, go into your original Web Catalog and merge it manually.

10 Save the Current Release Siebel Catalog.

This catalog contains the merged content.

Localizing Siebel Analytics Web Catalog Text Strings

For organizations that need to localize text strings in the Web Catalog, this section explains how to export the text strings for translation, and then expose them when translation is complete. You and your localization team are responsible for escaping characters properly and resolving any errors in the translated text strings.

The export process is based on Extensible Markup Language (XML). The export process creates one XML file for every first-level subfolder in the shared folder, in the format foldernameCaptions.xml, where foldername is the name of the subfolder in the shared folder. Each XML file contains the text strings for all content in the corresponding first-level folder and its subfolders.

For example, if the shared folder in the Web Catalog contains the first-level folders Marketing, Service, and Sales, then the export process creates three XML files:

- MarketingCaptions.xml
- ServiceCaptions.xml
- SalesCaptions.xml

After the content is translated, you will place these folders in the SiebelAnalyticsData directory. Their content is loaded when Siebel Analytics Web is started.

To export Web Catalog text strings

1 Stop the Analytics Web Server service.

2 Use Analytics Catalog Manager to open the Web Catalog that contains the text strings to be localized, and then choose Tools > Localization > Export.

3 Browse to select the location in which to write the output, and then click OK.

The export process may take several minutes.

4 Save the Web Catalog when prompted.

5 When the export process is complete, deliver the output file to your localization team.

You may need to make a copy of every output file for each language to be translated.

You can now restart the Analytics Web Server service and resume using Siebel Analytics Web.

To expose Web Catalog text strings

1 Place the translated XML files into their corresponding location in the SiebelAnalyticsData directory:

SiebelAnalyticsData\Web\Res\l_xx\Captions

where xx is the language extension, as shown in the following table.

Language Extension	Language
cs	Czech
da	Danish
de	German
en	English
es	Spanish
fi	Finnish
fr	French
it	Italian
ja	Japanese
ko	Korean
nl	Dutch
pt	Portuguese
pt-br	Brazilian Portuguese
sv	Swedish
zh	Chinese (Simplified)
zh-tw	Chinese (Traditional)

2 Restart the Analytics Web Server service to make the translated content available.

Best Practices for Working with the Siebel Analytics Web Catalog

This section gives best practices for working with Web Catalog content.

CAUTION: Make sure that the Analytics Catalog Manager is accessible only to Siebel Analytics Web administrators. As no login is required, full privileges for operations are granted to anyone who uses it.

- Use Analytics Catalog Manager only when Siebel Analytics is not running. Never make modifications to an in-use Web Catalog.

- Always make backups of the Web Catalogs that you are working with. Also back up the .version file.
- When you want to work with two Web Catalogs, always open the second catalog from the Analytics Catalog Manager instance that you used to open the first catalog.
- When you are copying or pasting content from one Web Catalog to another, you need to stay within the same relative structure. For example, you can copy a request from a User folder to another User folder, but not from a User folder to a Shared folder.
- Copying and pasting into email is not supported.
- Do not attempt to reorder items in the Web Catalog using Analytics Catalog Manager.

Viewing Information About Siebel Intelligence Dashboards

This section explains how to use Analytics Catalog Manager to view information about dashboards. Organizations with Siebel Analytics applications can locate and view information about preconfigured dashboards.

This section contains the following topics:

- "Exposing Dashboards and Requests in Siebel Analytics" on page 90
- "Locating Dashboard and Page Names in Siebel Analytics" on page 90

Exposing Dashboards and Requests in Siebel Analytics

Depending on the Analytics options your organization purchased, the Siebel Analytics Web administrator may need to expose them before the associated dashboards and requests can be viewed in Siebel Analytics Web and in Analytics Catalog Manager. This applies to sites that have the following Siebel Analytics applications and options:

- Siebel Sales Analytics, with the Forecasting and Incentive Compensation options.
- Siebel Service Analytics, with the Universal Queuing, Email Response, and Agreements options.
- Siebel Sales Analytics, with the Forecasting and Incentive Compensation options.
- Siebel Partner Analytics, with the Partner Marketing, Partner ERM, and Partner ISS options.

These options need to be exposed using the administration feature in Siebel Analytics Web. For instructions, see *Siebel Analytics Installation and Configuration Guide*.

Locating Dashboard and Page Names in Siebel Analytics

In Analytics Catalog Manager, the Web Catalog distributed with Siebel Analytics applications has the following structure:

Web Catalog > shared folder > Siebel Analytics application name > _Portal folder > dashboard name > dashboard page name

You can view the contents of a dashboard page in XML by clicking the Edit XML button when viewing the properties of a dashboard page.

CAUTION: If you change the XML code, you are changing the representation of the object in the Web Catalog.

Using Analytics Catalog Manager to Rename Items in the Siebel Analytics Web Catalog

You can use Analytics Catalog Manager to rename items in the Web Catalog. This can be useful when you are migrating from a development environment to a production environment.

In production environments, you may prefer to use the Manage Catalog screen in the Siebel Analytics Web user interface to rename items. Doing so allows you to preserve references that other users might have to the prior name of the item. References are not preserved when you use Analytics Catalog Manager to rename items. For more information about the Manage Catalog screen, see "Administering Items in the Siebel Analytics Web Catalog" on page 92.

Example Procedure: Renaming a Shared Siebel Intelligence Dashboard Using Analytics Catalog Manager

The following procedure provides the steps to rename a shared dashboard using Analytics Catalog Manager.

CAUTION: Do not rename the "My Dashboard" dashboards in the Users folder.

To rename a shared dashboard using Analytics Catalog Manager

1. Stop the Siebel Analytics Web Server service.
2. Make backup copies of your Web Catalog and the version file.

   For more information about the version file, see "How the Siebel Analytics Web Catalog Backup Process Works" on page 70.
3. Use Analytics Catalog Manager to open the Web Catalog and navigate to the dashboard you want to rename, for example, shared folder > _portal folder > dashboard name.
4. Right-click the dashboard name, choose Rename, and type a new name for the dashboard.
5. Save the Web Catalog.
6. Restart the Analytics Web Server service.

Administering Items in the Siebel Analytics Web Catalog

You administer shared Web Catalog folders through the Manage Catalog screen of Siebel Analytics Web. You can view folders and contents including hidden items, and create, rename, copy, move, and delete folders and contents. You can also take ownership of an item. (Taking ownership of an item allows only you to make changes to it. This is useful if a user needs assistance with, for example, a request.)

For information about changing permissions to an item in the Web Catalog, see "About Setting Siebel Analytics Web Permissions" on page 107.

To display the shared Web Catalog folders

1. In Siebel Analytics Web, do one of the following:
   - If you are running Siebel Analytics stand-alone, click the Admin link.
   - If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.
2. Click the link Manage Analytics Catalog.

   The Manage Catalog screen appears.
3. Click the Shared Folder link.

To show hidden items

- Select the Show Hidden Items check box.

  This allows you to see the following hidden items:
  - _Portal
  - _Filters
  - _Prefs
  - _Alerts
  - _iBots
  - _Delivers

  NOTE: The availability of the Show Hidden Items check box is controlled by the See Hidden Items privilege. For more information, see "Default Siebel Analytics Web Privilege Assignments" on page 114.

To create a new folder

- Click the link Create New Folder and follow the instructions on the screen.

To view the contents of a folder

- Click the name of the folder to see its content.

To take ownership of an item

1. Navigate to the item.
2. Click the Properties button for the item.

   The Item Properties screen appears.
3. If you have the appropriate authority, you can take ownership of the item or folder by clicking one of the following links:
   - Take Ownership of this item
   - Take Ownership of this item and all subitems

To rename a folder or an item

1. Navigate to the folder or item that you want to rename.
2. Click the Rename button.

   The Rename Item screen appears.
3. Type a new name for the item.

   You can also type a description.
4. To preserve references that other users might have to the prior name of the item, select the following option:

   Preserve references to the old name of this item

   If you do not select this option, users referencing this item are no longer able to see it.
5. Click the Rename button.

To delete a folder or an item

1. Navigate to the folder that you want to delete.
2. Click the Delete button for the folder.

   The Confirm Item Deletion screen appears.

   NOTE: When deleting shared folders or items, be aware that other users who have added shared items to their personal dashboards will no longer have access to them.
3. To confirm the delete, click Yes.

To copy or move an item

1. Navigate to the item.

2 Click Copy/Move button for the item.

The Copy/Move Item Here screen appears.

3 Follow the instructions given on the screen.

- To create a new folder, click the link Create New Folder and follow the instructions on the screen.

4 To show hidden items, select the Show Hidden Items check box.

This allows you to see the following hidden items:

- _Portal
- _Filters
- _Prefs
- _Alerts
- _iBots
- _Delivers

NOTE: The availability of the Show Hidden Items check box is controlled by the See Hidden Items privilege. For more information, see "Default Siebel Analytics Web Privilege Assignments" on page 114.

5 Click the Copy/Move Item Here button when you are done.

7 Managing Siebel Analytics Web Security

This chapter explains how to set up Siebel Analytics Web security so that users have only:

- Access to items in the Web Catalog that are appropriate to them.
- The ability to perform actions that are appropriate to them.

Use this chapter if your organization uses Siebel Analytics Stand-Alone or if your organization uses Siebel Analytics applications, but has created additional content outside of the applications.

This chapter contains the following topics:

Overview of Siebel Analytics Web Security

This section provides an overview of Siebel Analytics Web security. It contains the following topics:

Where Siebel Analytics Web Security Settings Are Made

Security settings that affect Siebel Analytics Web are made in the following Siebel Analytics components:

- **Server Administration Tool.** This is where you perform the following tasks:
  - Set permissions for business models, tables, columns, and subject areas.
  - Specify database access for each user.
  - Specify filters to limit the data accessible by users.
  - Set authentication options.

  For more information, see *Siebel Analytics Server Administration Guide*.
- **Siebel Analytics Web.** This is where you set permissions to Web Catalog items, including dashboards, and the privilege to perform actions such as edit views, create iBots, and create prompts.
- **Catalog Manager.** This is where you set permissions to Web Catalog items, including dashboards.

Your Security Goals in Siebel Analytics Web

Your main security goals are to make sure that:

- Only appropriate people can log on and access Siebel Analytics Web. This is achieved by assigning logon rights and authenticating users through the Siebel Analytics Server. For more information about authentication, see "About Siebel Analytics Web User Authentication" on page 102.
- Employees can access only the data that is appropriate to them. This is achieved by applying access control in the form of permissions.
- Employees have the ability to perform only actions that are appropriate to them. This is achieved by applying user rights in the form of privileges.

About Access Control and Permissions in Siebel Analytics Web

Access control defines the ability of an account to access a shared Web Catalog item. Catalog items are folders and requests, where folders are application folders, dashboard folders, and dashboard page folders.

An account is one of the following:

- An individual user.
- A Web Group that has one or more users as members.

Permissions describe the type of access to an object that an account is permitted. Examples are Read and Full Control.

Each Web Catalog item has an access control list that defines which accounts have which permissions to access the item. An access control list has the general form shown in Table 10.

Table 10. Access Control List for a Catalog Item

Account	Permission
Web Group 1	Read
Web Group 3	Full Control
Web Group 8	Read
User 4	Read
User 9	Full Control
User 11	Full Control

For more information about permissions, see "About Setting Siebel Analytics Web Permissions" on page 107.

About User Rights and Privileges in Siebel Analytics Web

Privileges are the actions that users have the right to perform in Siebel Analytics Web. Example privileges are "Edit system-wide column formats" and "Create iBots."

Privileges are managed by associating them with accounts, that is, individual users or Web Groups. A specific account is either granted or denied a specific privilege. These associations are created in privilege assignment tables.

The general form of a privilege assignment table is shown in Table 11 on page 97. The Web Groups in the right column are granted the privileges in the left column.

Table 11. Privilege Assignment Table

Privilege	Accounts That Have Been Granted the Privilege
Privilege 1	Web Group 2, Web Group 4
Privilege 2	Web Group1, Web Group 3
Privilege 3	Web Group 1, user 3
Privilege 4	Web Group 1, user 1, user 4, user 6
Privilege 5	Web Group 2, Web Group 3, user 4

For more information about privileges, see "About Setting Siebel Analytics Web Privileges" on page 113.

About User Authentication in Siebel Analytics Web

Authentication is the process of using a username and password to identify a someone who is logging on. Authenticated users are then given appropriate authorization to access a system, in this case Siebel Analytics Web. Siebel Analytics Web does not have its own authentication system; it relies on the authentication system built into the Siebel Analytics Server.

For more information about authentication, see "About Siebel Analytics Web User Authentication" on page 102.

Types of Siebel Analytics Web Groups

Web Groups are defined by the system or by a Siebel Analytics Web administrator. When a user is assigned to a Web Group, the user becomes a member of that group. Web Group membership is used to determine the permissions and privileges that are associated with a user, either by explicit assignment or inheritance.

Web Groups can also be thought of as roles for users because they avoid ambiguity about which defaults, preferences, and so on, to assign directly to the user.

This section contains the following topics:

System-Defined Web Groups in Siebel Analytics Web

System-defined Web Groups are preconfigured and required for successful Siebel Analytics Web operations. There are three types of system-defined Web Groups:

- **Everyone.** By default, all users belong to the Everyone group. This is why the group does not appear on the Groups and Users screen of the application.
- **Authenticated Users.** When a user is authenticated by the Siebel Analytics Server, the user automatically becomes a member of the Authenticated Users group. The Authenticated Users group is itself a member of the Everyone group.
- **Web Administrators.** Members of the Web Administrators group are users who are Siebel Analytics Web administrators. The default member of this group is the Siebel Analytics Web administrator. By default, only members of the Web Administrators group have access to administrative functions, but this can be changed by changing privilege assignments.

Administrator-Defined Web Groups in Siebel Analytics Web

Administrator-defined Web Groups are created by the Siebel Analytics Web administrator. You can create an unlimited number of Web Groups in Siebel Analytics Web.

NOTE: For organizations that have Siebel Analytics applications, preconfigured groups are set up with preconfigured responsibilities. For more information, see *Siebel Analytics Installation and Configuration Guide*.

The Web Groups that you define should be used to categorize users who require similar access to dashboards and content. You should plan your Web Catalog folder structure and Web Groups together to create a coherent security model. For more information about Web Catalog structure, see Chapter 6, "Administering the Siebel Analytics Web Catalog." For more information about how Web Group membership can be passed from the Siebel Analytics Server, see "About Siebel Analytics Web User Authentication" on page 102.

Administering Siebel Analytics Web Groups

When you create a Web Group, a group folder is automatically created in the Web Catalog. All members of the Web Group are given Read permissions to this folder.

To create a Web Group

1. In Siebel Analytics Web, do one of the following:
   - If you are running Siebel Analytics stand-alone, click the Admin link.
   - If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.
2. Click the following link:

   Manage Web Groups and Users
3. Under Existing Web Groups, scroll down and click the following link:

   Create a new Web Group

   The Create Web Group screen appears.
4. Enter a name for the group in the Group Name text box.

   The name of the Web Group must not match the name of any user who will log on to Siebel Analytics Web. This match is not case sensitive. For example, the user name FOO2 matches the Web Group named foo2 or FoO2. If a user and a Web Group share the same name, the user receives an Invalid Account message when attempting to log on to Siebel Analytics Web.

   NOTE: If the name of a Siebel Analytics Server group (which is set up in the Siebel Analytics Administration Tool) matches the name of a Web Group, members of the Siebel Analytics Server group automatically become members of the Web Group when they log on to Siebel Analytics Web. Their membership ceases when they log off. The name of the user never appears to a Siebel Analytics Web administrator as a member of the Web Group. The user appears only as a member of the Authenticated Users group. If you prefer to use Web Group names that match Siebel Analytics Server group names, but want users to be viewable as members of the group, with persistent group membership, you must explicitly add the users to the Web Group using the Web Security: Groups and Users screen.
5. (Optional.) Enter a password for the group in the Password text box, and type the password again in the Verify Password text box.

   Users are prompted for the password when they attempt to join the group.

6 In the Dashboard Name text box, type a name for the dashboard.

You can assign an existing dashboard to the Web Group or create a new dashboard for the Web Group. Whichever option you choose, all members of the Web Group are granted Read access to the designated, default dashboard. If you do not want to create a dashboard now, you can create one later from the Manage Dashboards screen.

If you do create a new, empty dashboard, it is created within a new, like-named shared folder. Group members will have Read permission to this folder.

7 In the Dashboard Builder box, specify the name of the user or group that has permission to change the contents of the dashboard. Separate multiple entries with a comma, for example, `user1,salesgroup`.

8 Click Finished.

NOTE: Siebel Analytics Web automatically gives a newly created Web Group Read permission to the automatically created group folder. For manually created group folders, you should set the permissions to the Group folder for the appropriate groups to Read.

To delete a Web Group

1 In Siebel Analytics Web, do one of the following:

- If you are running Siebel Analytics stand-alone, click the Admin link.
- If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.

2 Click the following link:

Manage Web Groups and Users

3 Under Existing Web Groups, in the Web Groups table, locate the group you want to delete and click the Delete button.

The Confirm Group Deletion screen appears.

4 Click Yes to confirm the delete.

5 In the Web Security: Groups and Users screen, click Finished.

NOTE: The procedure for deleting a Web Group does not delete any group definitions in a Siebel Analytics Repository.

To add a user or group to an existing group

1 In Siebel Analytics Web, do one of the following:

- If you are running Siebel Analytics stand-alone, click the Admin link.
- If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.

2 Click the following link:

Manage Web Groups and Users

3 Under Existing Web Groups, in the Web Groups table, locate the group whose membership you want to change.

4 Click the Edit button.

The Edit Web Group screen appears.

5 In the Group Membership area, click the link Add New Member.

The Add Member to Group screen appears.

6 In the Member Name text box, type the name of the user or group you want to add and click the Add Member button.

The Edit Web Group screen appears.

The Group Membership area now shows the list of the current members of the group.

7 Click Finished.

To delete a user or group from an existing group

1 In Siebel Analytics Web, do one of the following:

- If you are running Siebel Analytics stand-alone, click the Admin link.
- If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.

2 Click the following link:

Manage Web Groups and Users

3 Under Existing Web Groups, in the Web Groups table, locate the group whose membership you want to change and click the Edit button.

The Edit Web Group screen appears.

4 Under Group Membership, in the Members table, locate the member you want to delete and click the Remove button.

5 (Optional.) At the Edit Web Group screen, change the password for the group.

6 Click Finished.

NOTE: You cannot delete system-defined Web Groups or the user name Administrator.

To change the password for a group

1 In Siebel Analytics Web, do one of the following:

- If you are running Siebel Analytics stand-alone, click the Admin link.
- If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.

2 Click the following link:

Manage Web Groups and Users

3 Under Existing Web Groups, in the Web Groups table, locate the group whose password you want to change and click the Edit button.

   The Edit Web Group screen appears.

   The Password and Verify Password boxes are blank.

4 Under Group Properties, in the Password box, type the new password.

5 In the Verify Password box, type the password a second time to verify it.

6 Click Finished.

To delete a user from Siebel Analytics Web

1 Under Authenticated Web Users, in the Web User and Group Membership table, locate the user you want to delete and click the Delete button.

   The Confirm User Deletion screen appears.

2 Click Yes to confirm the delete.

3 In the Web Security: Groups and Users screen, click Finished.

NOTE: The procedure for deleting a user from Siebel Analytics Web does not delete any user definitions in a Siebel Analytics Repository. If the user logs on again, their Siebel Analytics Web entry is recreated.

About Siebel Analytics Web User Authentication

When users log on, they are authenticated by the Siebel Analytics Server using the authentication method specified in the Siebel Analytics Server configuration file NQSConfig.INI.

This section briefly describes the authentication methods that are relevant to Siebel Analytics Web users, in the context of Siebel Analytics Web. It contains the following topics:

For detailed information about authentication options, see *Siebel Analytics Server Administration Guide*.

Siebel Analytics LDAP or ADSI Authentication

If you are using LDAP or ADSI authentication to provide Siebel Analytics Server access control, you may also configure your LDAP or ADSI server to provide other security information. For example, you could configure the name that is displayed in the Welcome text when the user logs on to Siebel Analytics Web, and is specific to the Web Groups to which the user belongs. This information is contained in LDAP variables, which are passed to the Siebel Analytics Server session variables during the process of user authentication.

The variable USER is a system session variable that is used with LDAP or ADSI authentication. Whenever users log on to Siebel Analytics Web, their usernames and passwords are passed to the LDAP or ADSI server for authentication. After successful authentication, other system or non-system session variables for each user can be populated from information returned by the LDAP or ADSI server.

For more information about session variables, see "About Siebel Analytics Web Groups and Siebel Analytics Session Variables" on page 104.

Siebel Analytics External Table Authentication

If you are using an external database table for authentication, the table can contain additional access control information. This includes the name that is displayed in the Welcome text when the user logs on to Siebel Analytics Web, and the Web Groups to which the user belongs.

Whenever users log in, their usernames and passwords are authenticated through SQL that queries this database table for authentication. After successful authentication, the results of this SQL query can be used to populate other system and non-system session variables for each user.

For more information about session variables, see "About Siebel Analytics Web Groups and Siebel Analytics Session Variables" on page 104.

Siebel Analytics Database Authentication

The Siebel Analytics Server can authenticate users through database logons. When a user attempts to log on to Siebel Analytics Web, the Siebel Analytics Server attempts to use the logon name and password to connect to the authentication database, using the first connection pool associated with it. If this connection succeeds, the user is considered to be authenticated successfully.

Database authentication provides no mechanism to return additional access control information, such as the user's display name or Web Group membership. If you are using database authentication, alone or in conjunction with external table authentication, you need to explicitly add users to the appropriate Web Group.

NOTE: Siebel Delivers does not work with database authentication. For more information, see "About Siebel Delivers iBots and Impersonation" on page 49.

Siebel Analytics Internal Authentication

The Siebel Analytics Server internal authentication method has no way to return additional access control information. If you are using Siebel Analytics Server internal authentication, you need to explicitly add users to the appropriate Web Group, because the GROUP variable is not used with internal authentication.

About Siebel Analytics Web Groups and Siebel Analytics Session Variables

If the Siebel Analytics Server is using an external table or LDAP server for authentication, you must configure system session variables in the repository.

A session variable block contains the SQL statement that is issued when each session begins. This block can contain system session variables with fixed meanings (such as USER, GROUP, DISPLAYNAME, and WEBGROUP) and other non-system session variables unique to your particular environment. See *Siebel Analytics Server Administration Guide* for additional information about using session variables.

You should also create a Web group to match each possible value returned in the GROUP or WEBGROUPS variable for which you want to control privileges and permissions to Siebel Analytics Web components and requests.

This section contains the following topics:

About the Siebel Analytics GROUP Session Variable

The GROUP variable contains one or more group names, separated by semicolons, that are also used by the Siebel Analytics Server for security and content filtering. In many cases, these same groups are sufficient to control access to Siebel Analytics Web content. Siebel Analytics applications have been preconfigured to use this GROUP variable technique to inherit group memberships from the Analytics server.

About the Siebel Analytics WEBGROUPS Session Variable

The WEBGROUPS session variable allows greater flexibility because you can define Web Groups that categorize the roles, or classes, of Web users. For example, you might create the following groups:

- A Basic group that can only access the dashboard.
- A Standard group that has minimal access to Siebel Answers.
- A Power Users group that has full access to Siebel Answers and minimal access to iBots (Siebel Delivers).
- An Administrative group with full access to all features.

Setting Permissions and Privileges in Siebel Analytics Web

When you have set up the Web Groups, create a Web Catalog folder structure and assign appropriate privileges and permissions to each group. Keep in mind that each user can be associated with multiple roles by being a member of multiple Web Groups. Although WEBGROUPS can be used to control access to Web Catalog content (permissions), usually GROUP controls content and WEBGROUPS controls the ability to perform actions (privileges).

NOTE: Some GROUPs may not have corresponding Web content. In this case, when you create the group, you can delete the group folder created for the group in the /Shared folder, and give the group permission to the other group folders and subject area folders as appropriate (see Chapter 6, "Administering the Siebel Analytics Web Catalog" for more information).

For more information about the sequence in which to set up security, see "Guidelines for Configuring Siebel Analytics Web Security for the Web Catalog and Dashboards" on page 116.

For more information about permissions, see "About Setting Siebel Analytics Web Permissions" on page 107.

For more information about privileges, see "About Setting Siebel Analytics Web Privileges" on page 113.

Inheritance of Siebel Analytics Web Permissions and Privileges

Permissions and privileges can be assigned to users directly or through membership in groups. From another perspective, permissions and privileges can be assigned explicitly or effectively. Effective permissions and privileges are assigned indirectly through Web Group inheritance, which is the recommended way to set up your security. Permissions and privilege inheritance occurs when one Web Group is a member of another Web Group.

This section contains the following topics:

Rules for Inheritance in Siebel Analytics Web

- Any permissions or privileges given explicitly to a user override any permissions or privileges inherited from the Web Group to which the user belongs.

  As an example, All Authenticated Users have access to Dashboard X, except for George.

- If a user belongs to two groups and both groups are assigned permissions the least restrictive permissions are given to the user.

  For example, if one group allows Read access and another allows Change access, the least restrictive access would be granted; in this example, Change access.

  NOTE: The exception to this is if one of the two groups is explicitly denied the permissions, in which case the user is denied.

- If a user belongs to Web Group X, and Web Group X belongs to Web Group Y, any rule assigned to group X overrides any rule assigned to group Y.

  For example, if Marketing has Read permissions, Marketing Administrators, which is a member of Marketing, can have Full Control permissions.

- Explicitly denying access takes precedence over any other permissions or privileges.

When assigning permissions or privileges it often useful to look at resolved permissions for users and groups at the bottom of the screen to make sure that everyone is inheriting correctly.

Example of Inherited Privileges in Siebel Analytics Web

Figure 3 shows an example of how privileges are inherited through Web Groups.

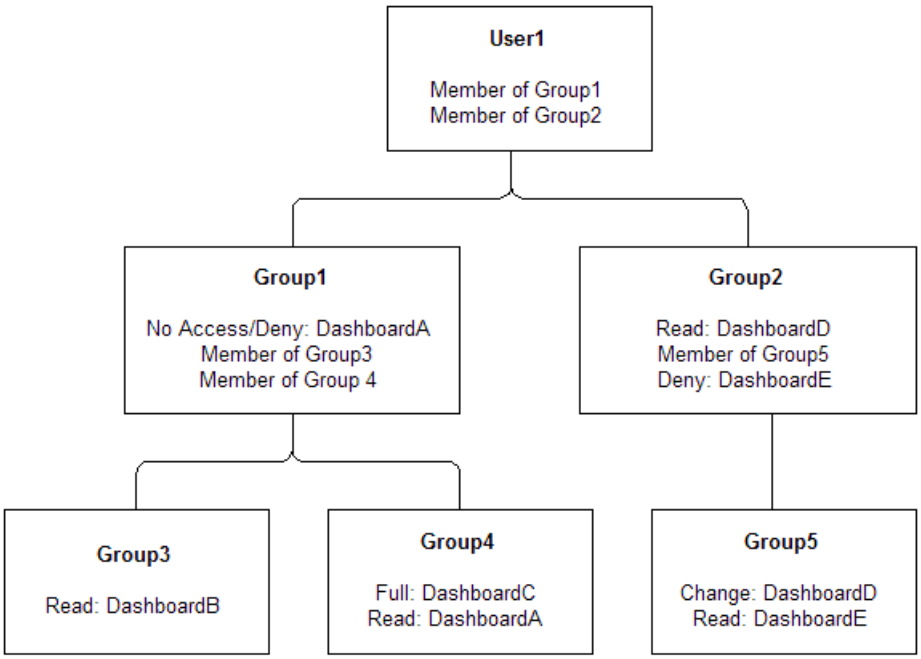


Figure 3. Example of Web Group Privilege Inheritance

In this example:

- User1 is a direct member of Group 1 and Group 2, and is an indirect member of Group 3, Group 4, and Group 5.
- The permissions and privileges from Group 1 are no access to DashboardA, Read access to DashboardB, and Full Control over DashboardC.
- If permissions and privileges are conflicting, the least restrictive level of authority is granted. Therefore, the inherited permissions and privileges from Group 2 include Change and Delete access to DashboardD.

- Specifically prohibiting access always takes precedence over any other settings. Therefore, Group 1's denial of access to DashboardA overrides Group 4's Read access. The result is that Group 1 provides no access to DashboardA. Likewise, Group 5 provides no access to DashboardE because access to it is explicitly denied in Group2.

The total permissions and privileges granted to User1 are as follows:

- No access to DashboardA and DashboardE because access is specifically denied.
- Read access to DashboardB.
- Full Control over DashboardC.
- Change and Delete access to DashboardD.

TIP: It is recommended that you do not add the default Everyone or Authenticated Users Web Groups to any other Web Groups that you create. This makes sure that only the desired Web Groups (and users) have the specified permissions and privileges, by preventing users or authenticated users from unintentionally inheriting permissions and privileges from another Web Group.

About Setting Siebel Analytics Web Permissions

Permissions are used to control access to shared information contained in:

- Siebel Analytics Web Catalog items
- Siebel Intelligence Dashboards

Permissions, which may be explicitly set or inherited, are configured from:

- Siebel Analytics Web Administration screen
- Analytics Catalog Manager

  To set permissions in the Web Catalog using Analytics Catalog Manager, Siebel Analytics Web must not be running. Use this option when you are working in a development environment or when you plan to make many changes. You can set permissions from the level of the application all the way down to individual requests in either Siebel Analytics Web or Siebel Analytics Web Catalog.

This section contains the following topics:

- "Types of Permissions in Siebel Analytics Web" on page 107
- "Recommendations for Setting Permissions in Siebel Analytics Web" on page 108

Types of Permissions in Siebel Analytics Web

Siebel Analytics Web supports four permissions:

- **No Access.** Access is not allowed for this user or group. Explicitly denying access takes precedence over any other permission.

- **Read.** Authority given to view the contents of the item, folder or dashboard, but cannot make any changes.
- **Change/Delete.** Authority given to view content, and make changes or delete the content.
- **Full Control.** Authority given to view content, make changes or delete the content, set permissions, and delete the item, folder or dashboard.

Recommendations for Setting Permissions in Siebel Analytics Web

Follow these recommendations when setting permissions:

- Assign permissions through Web Group membership, even if you want to assign permissions for a single user. For more information, see "Types of Siebel Analytics Web Groups" on page 98.
- Set the permission to the Group folder for the appropriate groups to Read.
- For groups (or users, if necessary) that are going to be modifying the dashboards and dashboard content accessible to the group, set the permissions for the group to Full Control. This is often a dashboard or content builder group. While allowing change and delete control, Full Control also allows the specified group (or user) to set permissions, and to delete the item, folder, or dashboard.
- For each Subject Area, grant Read permissions to the corresponding Subject Area folder within the Requests folder (and everything it contains). Make sure that the Authenticated Users and Everyone groups have no access effective permission to the Subject Area folder.
- For groups that should be able to save requests for public use against a given Subject Area, grant them Full Control to the Subject Area folder and everything it contains, and likewise for the Common folder. See the Change Item Permissions help for details on managing permissions for folders, items, and dashboards.
- To make sure that only members of the designated Web Groups (or users) have access to Siebel Analytics Web Catalog folders, folder content, and Siebel Intelligence Dashboards, do not set explicit permissions for the default Web Groups Authenticated Users or Everyone.

NOTE: Siebel Analytics Web does not allow you to remove permissions for yourself or for the administrator. This prevents you from locking yourself out of an item, folder, or dashboard.

TIP: If you want to provide a place for all users within a group to share requests with each other, create a folder under the Subject Area folder called, for example, Share or Publish, and give the entire group Change/Delete permission to just that folder.

Overview of the Siebel Analytics Web Administration Screen

The Siebel Analytics Web Administration screen contains the following sections:

- **Product Information.** This section provides information about your current installation, such as the product version in use, the path to the current Web Catalog, and a link to a list of features licensed by your organization.

- **Activities.** This section provides links to administrative functions. Table 12 lists and describes the links in the Activities section.

NOTE: The Activities section may also contain links for other Siebel applications that your organization has licensed, such as Siebel onDemand or Siebel Marketing. Such links are not listed in Table 12. For information about an application-specific link, click the Help button (if present) on the page that opens when you click the link, or consult the administrator documentation for the application.

Table 12. Description of Links in the Activities Section in Siebel Analytics Web Administration

Link	Description
Manage Web Groups and Users	Opens the Web Security: Groups and Users page, where you can control access to Siebel Analytics Web for both Web groups and individual users.
Manage Analytics Catalog	Opens the Manage Catalog page, where you can edit, rename, set permissions for, and delete folders and items in the Catalog. This feature can also be accessed by clicking the Manage button in the left pane in Siebel Answers.
Manage Intelligence Dashboards	Opens the Manage Dashboards page, where you can administer dashboard security. Depending on the level of authority granted to you based on the user name you log on with, you can use this page to access Web security, create and delete dashboards, and change user and group permissions to dashboards.
Manage Sessions	Opens the Session Management page, where you can manage active sessions, such as canceling running requests and viewing the log file NQQuery.log for information about a request.
Manage iBot Sessions	Opens the iBot Session Management page, where you can view information about currently active iBot sessions. For information, see "Viewing Information About Active Siebel Delivers iBot Sessions" on page 53.
Manage Privileges	Opens the Privilege Administration page, where you can manage privileges and rights for both Web groups and individual users to various Siebel Analytics Web features and functions.
Issue SQL	Opens the Issue SQL Directly page, where you can enter an SQL statement to be issued directly to the Siebel Analytics Server. This feature is useful for testing the server only. Results are not formatted, and it is not possible to save the SQL issued here as a request.
Reload Files and Metadata	Reloads XML message files, refreshes server metadata, and clears Siebel Analytics Web caches.

Setting Siebel Analytics Web Permissions Through the Administration Screen

You can change permissions to Web Catalog items through the Administration screen of Siebel Analytics Web.

To change permissions for a shared dashboard

1. In Siebel Analytics Web, do one of the following:
   - If you are running Siebel Analytics stand-alone, click the Admin link.
   - If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.
2. Click the following link:

   Manage Intelligence Dashboards
3. In the row of the dashboard for which you want to change permissions, click the Permissions button.

   The Change Item Permissions screen appears.
4. In the access control list, select a user or group, and click the corresponding link to toggle through the types of permissions.
5. Click the following link, if appropriate:

   Replace permissions with parent folder's permissions
6. Select one or both of the following options, as appropriate:
   - Apply permissions to subfolders
   - Apply permissions to items within folder
7. Click Finished.
8. In the Manage Dashboards screen, click Finished.

To change permissions for a dashboard page

1. In Siebel Analytics Web, do one of the following:
   - If you are running Siebel Analytics stand-alone, click the Admin link.
   - If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.
2. Click the following link:

   Manage Intelligence Dashboards

3 In the row of the dashboard for which you want to change permissions, click the Properties button.

4 In the Existing Pages area, identify the page and click the associated Security button.

5 In the access control list, select a user or group, and click the corresponding link to toggle through the types of permissions.

6 Click the following link, if appropriate:

Replace permissions with parent folder's permissions

7 Select one or both of the following options, as appropriate:

- Apply permissions to subfolders
- Apply permissions to items within folder

8 Click Finished, and then click either Yes or Yes All to confirm the override of permissions.

9 In the Dashboard Properties page, click Finished.

10 In the Manage Dashboards page, click Finished.

To change permissions for a shared Web Catalog item

1 In Siebel Analytics Web, do one of the following:

- If you are running Siebel Analytics stand-alone, click the Admin link.
- If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.

2 Click the link Manage Analytics Catalog.

3 Click the Shared Folder link to make sure you are working with shared folders.

4 Click a folder link and drill down to the relevant item.

5 In the row of the relevant Web Catalog item, click the Permissions button.

The access control list for the item appears.

6 Identify the user or group that has explicit access to this item and whose permission you want to change.

7 Click the link next to the group or user until the permission that you want to assign appears.

8 Click Finished.

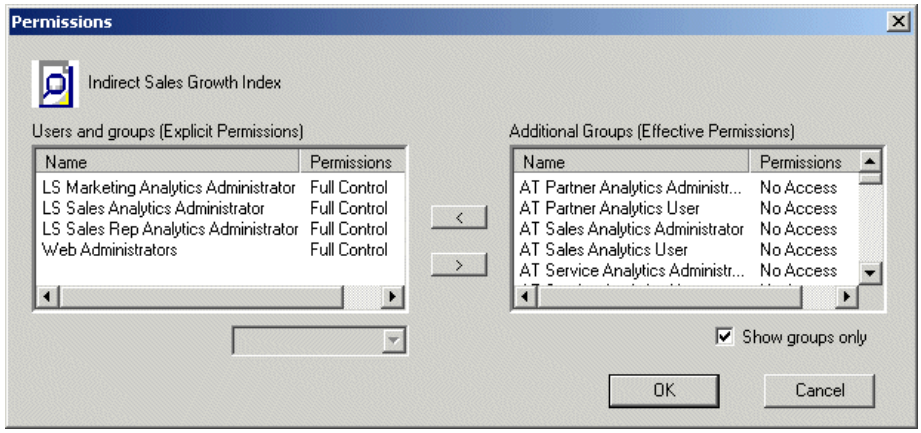
9 In the Confirm Override of Permissions section, click Yes to confirm.

Setting Siebel Analytics Web Permissions Through Analytics Catalog Manager

You can change permissions to Web Catalog items through Analytics Catalog Manager. For general information about Analytics Catalog Manager, see Chapter 6, "Administering the Siebel Analytics Web Catalog."

To set permissions

1. Open Analytics Catalog Manager (see "About Siebel Analytics Catalog Manager" on page 75).
2. Navigate to an item.
3. To review permissions for the item, right-click the item in the Name list and select Permissions.



The Explicit Permissions list on the left side of the window shows the users and groups that have explicit permissions granted to this item. The Effective Permissions list on the right side of the window shows the users and groups that have access granted though group inheritance, and users and groups that have no access to the request.

Permission	What It Means
R	Read permission.
W	Write permission
X	Traverse Folder permission. You can view the folders within this folder.
D	Delete permission.
P	Change Permission authority. You can alter the permissions on the item.
O	Owner permission, also known as Full Control. This includes the permissions in this table and means that you have full control over the item. This is the level of authority granted by default to the preconfigured Administrator user ID.

4 If you need to move a user or group into the Explicit Permissions list, select the name in the Effective Permissions list and click the left arrow button (<).

NOTE: To change a permission for user or group, it must be in the Explicit Permissions list. To view groups only, select the option Show groups only.

5 Select a user or group in the Explicit Permissions list.

6 Select the new permission from the drop-down list following the Explicit Permissions list.

7 Click OK to accept the changes.

8 Close the Permissions window.

NOTE: If you move a user or group from the Explicit Permissions list to the Effective Permissions list, the user or group privileges are reset to No Access. To move users or groups from one window to another, highlight them and click the right or left arrow button as appropriate.

About Setting Siebel Analytics Web Privileges

Privileges are the actions that users have the right to perform in Siebel Analytics Web. They are managed by associating them with accounts, that is, individual users or Web Groups. A specific account is either granted or denied a specific privilege. These associations are created in privilege assignment tables.

Like permissions, privileges are either explicitly set or inherited through group membership. Explicitly denying a privilege takes precedence over any granted, inherited privilege. For example, if a user is explicitly denied access to the privilege to edit column formulas in Siebel Answers, but is a member of a group that has inherited the privilege, the user cannot edit column formulas.

Unlike permissions, privileges are more commonly granted to the Everyone system Web Group. This allows users access to common Siebel Analytics Web features and functions.

For a list of some Siebel Analytics Web privileges, see "Default Siebel Analytics Web Privilege Assignments" on page 114.

Setting Siebel Analytics Web Privileges

The following procedures are used to administer privileges.

To assign a user or group to a privilege

1 In Siebel Analytics Web, do one of the following:

- If you are running Siebel Analytics stand-alone, click the Admin link.
- If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.

2 Click the Manage Privileges link.

3 Locate the privilege you want to change.

4 Click the user or Web Group name on the same row as the privilege.

The Privilege screen appears.

The screen contains two tables. The top table lists Web Groups and users with explicitly assigned privileges. The bottom table lists Web groups with inherited access (effective permissions) to this privilege.

5 (Optional.) To toggle the display to include users with inherited access, click the following link:

Show users and groups

6 In the bottom table, identify the group or user to allow explicit access and click the Add link on the same row.

The Web group or user is added to the top table.

NOTE: By default, a user who creates an item has the permission to change it. However, in some cases, this privilege should be disabled. Make the appropriate changes to deny the target user or group access to the Admin: Catalog Change Permissions privilege.

To change explicit access to a privilege

1 In Siebel Analytics Web, do one of the following:

- If you are running Siebel Analytics stand-alone, click the Admin link.
- If you are using a Siebel Analytics application, choose View > Site Map and access Analytics Administration.

2 Click the Manage Privileges link.

3 Locate the privilege you want to change in the upper table.

4 Do one of the following:

- Click the link to toggle between Granted and Denied.
- Click the Remove Permissions button to delete explicit access to the privilege. In this case, the Web group or user moves to the bottom table, displaying inherited permissions.

5 Click Finished.

Default Siebel Analytics Web Privilege Assignments

Table 13 lists some privileges that can be controlled, along with the Web Group that is granted access to that privilege by default.

These privileges apply to Siebel Analytics Stand-Alone only. If you are using Siebel Analytics applications, see *Siebel Analytics Installation and Configuration Guide* for information about preconfigured privileges, called responsibilities.

NOTE: The Privilege Administration screen may also contain privileges for other Siebel applications that your organization has licensed, such as Siebel Marketing. Web groups granted permission to application-specific components have descriptive names, such as Marketing Analytics User. For information about application-specific privileges, see the administrator documentation for the application.

Table 13. Privileges and Default Settings for Siebel Analytics Stand-Alone

Component	Privilege Description	Web Group Granted Permission
Access	Access to Siebel Answers	Everyone
	Access to Siebel Delivers and iBots	Everyone
	Access to Siebel Analytics Web Administration	Web Administrators
Admin: Catalog	Change Permissions	Everyone
Admin: General	Manage Sessions	Web Administrator
	Manage Dashboards	Web Administrators
Admin: Security	Manage Web Groups and Users	Web Administrators
	Manage Privileges	Web Administrators
	Take Ownership of Catalog Objects	Web Administrators
Catalog	Personal Storage (My Folders and My Dashboard)	Everyone
Formatting	Edit system-wide column formats	Web Administrators
	Edit personal column formats	No Access
Siebel Answers	Create Views	Everyone
	Create Prompts	Everyone
	Access the Advanced tab	Everyone
	Edit Column formulas	Everyone
	Edit Column filters	Everyone
	Advanced SQL	Everyone

Table 13. Privileges and Default Settings for Siebel Analytics Stand-Alone

Component	Privilege Description	Web Group Granted Permission
Siebel Delivers and iBots	Retrieve delivery destinations for iBots (system call)	Web Administrators
	Create iBots	Everyone
	Publish iBots for subscription	Everyone
	Deliver iBots to specific or dynamically determined users	Web Administrators
	Chain iBots together	Everyone
	Chain iBots to custom scripts	Web Administrators
Subject Area (by its name)	Access within Siebel Answers	Everyone
View: Chart	Add and Edit Chart View	Everyone
View: Column Filter	Add and Edit Column Filter View	Everyone
View: Compound	Add and Edit Compound View	Everyone
View: Filters	Add and Edit Filters View	Everyone
View: Global Filters	Add and Edit Global Filter View	Everyone
View: Image	Add and Edit Image View	Everyone
View: Logical SQL	Add and Edit Logical SQL View	Everyone
View: Narrative	Add and Edit Narrative View	Everyone
View: Pivot Table	Add and Edit Pivot Table	Everyone
View: Question	Add and Edit Question View	Everyone
View: Static Text	Add and Edit Static Text View	Everyone
View: Table	Add and Edit Table View	Everyone
View: Ticker	Add and Edit Ticker View	Everyone
View: Title	Add and Edit Title View	Everyone

Guidelines for Configuring Siebel Analytics Web Security for the Web Catalog and Dashboards

To set up a secure Web Catalog and secure dashboards, you must understand the information presented in the previous chapters, because you will be working with Web Catalog, dashboards, and Web Groups.

This section contains the following topics:

Creating Siebel Analytics Web Groups

When you create a Web Group, a shared Web Catalog group folder is automatically created. For more information about Web Groups, see "Types of Siebel Analytics Web Groups" on page 98. For more information about the group folder, see Setting Up the Siebel Analytics Web Catalog Structure.

Setting Up the Siebel Analytics Web Catalog Structure

The Web Catalog has two main folders:

- **/Shared.** Contains the personal storage and My Dashboards for each user.
- **/User.** Contains shared folders, shared dashboards, and shared dashboard content.

Figure 4 shows the recommended higher-level folder structure for the Web Catalog.

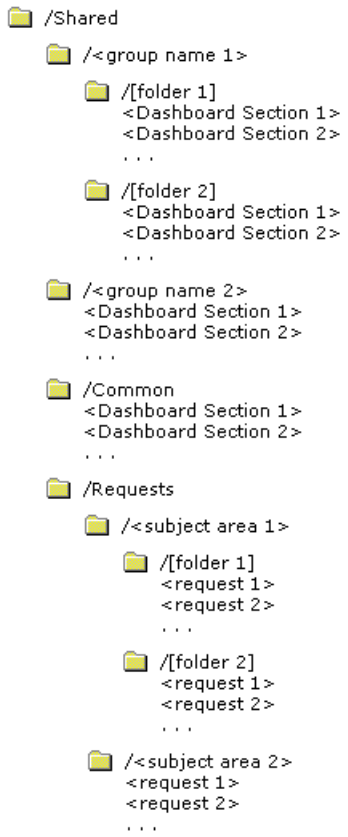


Figure 4. Recommended Catalog Folder Structure

The Web Catalog structure shown in Figure 4 makes it easier for users and administrators to reorganize content and make shared dashboard sections available. This, in turn, facilitates My Dashboard creation by allowing users to select from shared content.

Items shown within brackets ([ ]) are optional. Items shown within angle brackets (<) should be replaced with the appropriate name.

Group Folder

When you create a Web Group, the system automatically creates a shared folder for the group. All members of the Web Group are automatically given Read permissions to this folder. These permissions are inherited by any subfolders that you create.

Group folders hold shared dashboard content that pertains to only members of the Web Group. The shared content can be further organized into subfolders under the Group folder.

NOTE: You may want to remove this folder and grant group permissions to other folders if there is no group-specific content.

Common Folder

Dashboard sections that are shared across groups are saved in a Common folder under the /Shared folder. Content can be organized into subfolders under the Common folder.

Requests Folder

When you create requests in Siebel Answers that you want to share, save them in a Requests folder in the /Shared folder.

Create a folder for each subject area in the Requests folder for storing requests against that subject area. This allows you to manage permissions at the level of the subject area. If two or more subject areas are closely related, so that users with permission to one would almost always have permission to the other, you could create a single folder to store requests against both subject areas.

NOTE: If a user is given permission to a request in the Web Catalog that references a Subject Area to which the user does not have permission, the Siebel Analytics Server still prevents the user from executing the request.

In cases where you have requests that span subject areas (requests that involve SQL subqueries), you can put them in the folder for any of the subject areas, or create a new folder for users that have permission to these subject areas.

For more information about working with the Web Catalog, see Chapter 6, "Administering the Siebel Analytics Web Catalog."

Setting Permissions to Siebel Analytics Web Catalog Items

Before setting permissions to the Web Catalog items you have created, review the information in "About Setting Siebel Analytics Web Permissions" on page 107.

Creating Shared Siebel Intelligence Dashboards

After setting up the Web Catalog structure and setting permissions, you can create shared dashboards and content for reuse by others. If you did not specify a dashboard when creating a group, you can create one now, selecting the appropriate group folder.

The advantage to creating shared dashboards is that sections that are created in the shared dashboard are actually shortcuts to folders in the /Shared folder. As such, you can remove them and add them again in a different column, page, or even a different dashboard. Users can create a My Dashboard from existing shared sections by clicking the Add Folder link and selecting the appropriate folder from the /Shared/Group or /Shared/Common folders in the Web Catalog.

If you plan to allow multiple users to modify the Web Group's default dashboard, consider putting these users into another group. For example, suppose you create a Web Group called Sales and create a default dashboard called SalesHome. Of the 40 users that are members of the Sales group, suppose that there are three who need to have the ability to create and modify content for the SalesHome dashboard. It is recommended that you create a SalesAdmin group, with the same permissions of the primary Sales group. Add the three users who are allowed to make changes to the SalesHome dashboard and content to this new SalesAdmin group, and give this group the appropriate permissions to the Web Catalog. This allows them to create and modify content for the SalesHome dashboard. If a user no longer requires the ability to modify dashboard content, you can change the user's group membership to Sales. If an existing Sales group member needs to have the ability to create dashboard content, the user's group membership can be changed to SalesAdmin.

Dashboards are stored in a special folder, _Portal, that is hidden (unless the option to show hidden items is checked when working with the Web Catalog). The hidden _Portal folder is located directly underneath a group folder. Because the dashboard is contained in the Group folder, all the permissions you set to the group folder are inherited.

You can create multiple dashboards within a group folder, each of which has its own folder in the /_Portal folder, named from the dashboard name that you assign.

Underneath the specific dashboard folder, for example, /_Portal/SalesGroup Dashboard, are the pages for the dashboard.

For more information about creating shared dashboards, see "Creating and Deleting Shared Siebel Intelligence Dashboards" on page 55.

Adding Shared Siebel Intelligence Dashboards Pages and Content

After you have created dashboards, you can add pages and content.

Adding Pages and Content

For information about adding pages and columns, see *Siebel Analytics User Guide*.

Adding Sections

A section is a folder, or a shortcut to a folder, that appears within a dashboard. For the procedure for adding sections to a dashboard, see "Creating and Deleting Shared Siebel Intelligence Dashboards" on page 55.

When adding requests from Siebel Answers to a section, do one of the following:

- Include requests previously saved to the Subject Area folders.
- Create a new request, save it in the appropriate Subject Area folder, and add it to the shared section using the Existing Request link.

  This technique is preferred for several reasons. The permissions on the Subject Area folders automatically filter requests from the dashboard from users that might have dashboard permission, but not permission to certain subject areas. It is easy to reference the same request in multiple dashboard sections, you can change it once, and have it reflected in all sections.

If you are working with content that is specific to a number of group folders, you might want to create a new folder directly underneath the /Shared folder to use, and set permissions to the new folder to Read for the appropriate groups.

Testing the Siebel Intelligence Dashboards

Before releasing dashboards and content to the user community, perform some tests.

To test the dashboard

1. Verify that users with appropriate permissions can correctly access it and view the intended content.
2. Verify that users without appropriate permissions cannot access the dashboard.
3. Verify that styles and skins are displayed as expected, and that other visual elements are as expected.
4. Correct any problems you find and test again, repeating this process until you are satisfied with the results.

Setting Up a Virtual Directory for Shared Siebel Analytics Documents

It is recommended that you set up a virtual directory on the Web server for shared documents. Name the directory /DashboardFiles and map it to a shared network directory of the same name.

This allows users with the appropriate permissions to publish files to this folder, and reference these files by their relative URL names rather than by their fully qualified network share names, for example, `/DashboardFiles/AnnualReport.doc` instead of `\\SharedServer\CommonShare\DashboardFiles\AnnualReport.doc`.

Releasing Siebel Intelligence Dashboards to the User Community

After testing is complete, notify the user community that the dashboard is available, providing the relevant network address.

8 Using Siebel Analytics Web Logging

Siebel Analytics Web includes a logging facility that logs information that can be used to troubleshoot problems. This logging facility is also highly configurable. The purpose of this Technical Note is to describe the configuration parameters for the logging facility, and to provide information about Siebel Analytics Web Server log files.

This chapter contains the following topics:

See also Technical Note 519 (about interpreting Siebel Analytics log information) available on Siebel SupportWeb.

Using the Siebel Analytics Web Logging Facility

By default, Siebel Analytics Web is configured to log all error events and informational and warning events of sufficient importance. An example of an important informational event would be a server starting up or a server shutting down.

In order to debug specific issues that a user may be encountering, the logging level can be increased to log more information than the default configuration. For example, while debugging a particular Siebel Analytics Web Server connectivity issue, it might be useful to turn up maximum logging on the saw.odbc log source only. This allows you to add detailed logging for that component, without cluttering the log with detailed logging from other events in the system. Another example is to create a new log writer that records only chart events. (See Figure 7 on page 126 for an illustration of this example.)

CAUTION: Logging should not be turned up on a production system, except to diagnose specific issues. Logging does have an effect on performance.

All Siebel Analytics Web configuration information is loaded from the file `logconfig.xml`, located in the following directory (based on operating system platform):

- Windows:

  `SiebelAnalyticsData/web/config`

- UNIX:

  `INSTALLDIR/Data/web/config`

Analytics Web Configuration File Structure

The structure of the configuration XML file is shown in Figure 5 on page 124. The cardinality of each node is shown in brackets.

```
Config
Default  [1..1]
    Writers  [0..1]
        Writer  [0..1]
            Filters  [0..1]
                FilterRecord  [0..n]
    WriterClassGroups  [0..1]
        WriterClassGroup  [0..n]
    Filters  [0..1]
        FilterRecord  [0..n]
```

Figure 5. Structure of a logconfig.xml File

An example of a `logconfig.xml` file that has four writers is shown in Figure 6 on page 125.

```
<?xml version="1.0" ?>
<Config>
    <Default>
    <Writers>
        <Writer implementation="FileLogWriter" name="Global File Logger"
          writerClassId="1" dir="{%SADATADIR%}/web/log" filePrefix="sawlog"
          maxFileSizeKb="10000" filesN="10" />
        <Writer implementation="CoutWriter" name="Global Output Logger"
          writerClassId="2" />
        <Writer implementation="EventLogWriter" name="Event Logger"
          writerClassId="3" />
        <Writer implementation="CrashWriter" name="CrashWriter" writerClassId="4"
        />
    </Writers>
    <WriterClassGroups>
        <WriterClassGroup name="All">1,2,3,4</WriterClassGroup>
        <WriterClassGroup name="File">1</WriterClassGroup>
        <WriterClassGroup name="Cout">2</WriterClassGroup>
        <WriterClassGroup name="EventLog">3</WriterClassGroup>
        <WriterClassGroup name="Crash">4</WriterClassGroup>
    </WriterClassGroups>
    <Filters>
        <FilterRecord writerClassGroup="Cout" path="saw" information="31"
          warning="41" error="41" security="41" />
        <FilterRecord writerClassGroup="File" path="saw" information="31"
          warning="100" error="100" security="41" />
        <FilterRecord writerClassGroup="File" path="saw.mktgsqlsubsystem.joblog"
          information="41" warning="100" error="100" security="41" />
        <FilterRecord writerClassGroup="File" path="saw.httpserver.request"
          information="51" warning="100" error="100" security="41" />
        <FilterRecord writerClassGroup="File" path="saw.httpserver.response"
          information="51" warning="100" error="100" security="41" />
    </Filters>
    </Default>
</Config>
```

Figure 6. Example of a logconfig.xml File with Four Writers

An example of a `logconfig.xml` file that has one writer set up to record charting events is shown in Figure 7 on page 126.

```
<Config>
<Default>
      <Writers>
            <Writer implementation="FileLogWriter" name="Global File Logger" writerClassId="1"
                  dir="{%SADATADIR%}/web/log" filePrefix="sawlog" maxFileSizeKb="10000"
                  filesN="10" />
            <Writer implementation="CoutWriter" name="Global Output  Logger"
            writerClassId="2"/>
            <Writer implementation="EventLogWriter" name="Event  Logger" writerClassId="3" />

        <!-- New log writer dedicated for charts -->
            <Writer implementation="FileLogWriter" name="Chart Logger"
            writerClassId="4"
                  dir="{%SADATADIR%}/web/log/chart" filePrefix="sawlog"
                  maxFileSizeKb="10000" filesN="10" />
      </Writers>
      <WriterClassGroups>
            <WriterClassGroup name="All">1,2,3,4</WriterClassGroup>
            <WriterClassGroup name="File">1</WriterClassGroup>
            <WriterClassGroup name="Cout">2</WriterClassGroup>
            <WriterClassGroup name="EventLog">3</WriterClassGroup>
            <WriterClassGroup name="Chart">4</WriterClassGroup>
      </WriterClassGroups>
      <Filters>
            <FilterRecord writerClassGroup="Cout" path = "saw" information="31" warning="41"
            error="41" security="41"/>
            <FilterRecord writerClassGroup="File" path = "saw" information="31" warning="100"
            error="100" security="41"/>
            <FilterRecord writerClassGroup="File" path = "saw.mktgsqlsubsystem.joblog"
            information="41" warning="100" error="100" security="41"/>
        <!-- Logs all chart events, including minor informational events -->
            <FilterRecord writerClassGroup="Chart" path = "saw.charts"
            information="100" warning="100" error="100" security="100"/>
            <FilterRecord writerClassGroup="Chart" path = "saw.views.chart"
            information="100" warning="100" error="100" security="100"/>
        </Filters>
</Default>
</Config>
```

Figure 7. Example of a logconfig.xml File with ChartLogger Writer

A description of each node in the configuration hierarchy is shown in Table 14 on page 127.

Table 14. Siebel Analytics Web Log Configuration File Elements

Element	Attribute	Description
Writers	Contains writers configuration. This configuration is loaded on startup.	
Writer	Configures a writer.	
	implementation	Name of the C++ class that implements the writer. The following implementations are defined: ■ FileLogWriter. Writes to a disk file. ■ CoutWriter. Writes to standard output. ■ EventLogWriter. Writes to Windows event log or UNIX syslog. ■ CrashWriter. Writes to a crash dump file when the Analytics Web Server attempts to log from a specific source file and line number. ■ Used in a production environment for information of some loggable but non-fatal error (for example, failed NQTEST). ■ On Windows, CrashWriter requires appropriate version of `dbghelp.dll` (at least 6.0.17.0). The correct `dbghelp.dll` is found in `support/windows/system32`. Put this DLL in the `WINNT/system32` or in the `main/bin` directory. No registration is required.
	name	Unique name for the writer.
	writerClassId	Integer number in the range 1-10. This number is used by filters to allow or prohibit logging. Each distinct writer must have a unique value, which is used later for filter configuration. Different writers may have the same class ID, but if they do, those writers cannot be distinguished by filters.

Table 14. Siebel Analytics Web Log Configuration File Elements

Element	Attribute	Description
Writer (continued)	FileLogWriter specific attributes:	
	dir	Directory where log files are to be created.
	maxFileSizeKb	Maximum size of the logging file in kilobytes. When the file size limit is reached, the file is closed and a new logging file is created.
	filePrefix	Log files prefix.
	filesN	Maximum number of logging files. When this number is exceeded, the logger starts to write to the beginning of the first file.
	EventLogWriter specific attributes:	
	winSource	Event log source for logged events.
	CrashWriter specific attributes:	
	file	Dump file path. On Windows, a dump file is created in bin/coredumps and Siebel Analytics Web Server continues to run.
	line	Dump file line number.
WriterClassGroups	Contains definition for writer classes. Writer class is a group of Writer class IDs.	
WriterClassGroup	Contains (as child text) a comma-separated list of class IDs.	
	name	Name of the WriterClassGroup.
Filters	Contain filter configuration.	
FilterRecord	writerClassGroup	Specifies the group of writers to which this record is applied. WriterClassGroup should be defined previously in the WriterClassGroups section.
	path	Log source path. Current filter record is applied to the software component identified by that path and all its subcomponents.
	information	An integer that specifies severity for corresponding message type. Only messages with a severity index less than the provided number are logged.
	warning	
	error	
	security	

Analytics Web Message Structure

Each message that is logged by Siebel Analytics Web has several components to it, as shown in Table 15 on page 129.

Table 15. Siebel Analytics Web Log Message Components

Message Component	Description
Message Text	The text of the log message to the user.
Message Type	One of four types: information, warning, error and security. The first three are self-explanatory. Security is reserved for auditing security type of information such as user logged in, login failed, user accessed catalog item XYZ, and so on.
Severity	The severity is represented as a nonnegative integer. The lower the value, the more important the message. A message with severity 0 is the most important type of message; a message with severity 1000 is not important at all.
Log Sources	Log sources indicate where the message is coming from. Sources are always of the form saw.component.subComponent.function. There is no limit on the number of "dots" in a source. The source can be as many levels deep as the programmer decides to make it. Moreover, each logged message may have one or more log sources associated with it, depending on what code path caused that error to be logged. For example, the message "Unable to open file" could be logged with the stack of sources *{saw.delivers, saw.charts}* as well as the log source stack *{saw.views, saw.pdf}*. The command option sawserver `/logsources` (case insensitive) prints all known log sources.
Message Properties	Properties that indicate various other kinds of information. The kind varies from message to message, and may include username, IP address of client browser, thread ID, and so on.

Analytics Web Logging Levels

The categories, impact, and descriptions of the log levels are shown in Table 16 on page 130.

Table 16. Siebel Analytics Web Logging Levels

Category	Level	Impact	Description
Errors	10	Corruption	Data corruption detected.
	20	Fatal	Cannot recover without restart.
	25	Unknown	Special severity case for catch (...).
	30	Critical	A recoverable error that needs attention.
	40	Error	Basic error message.
	45	User	Special severity for user input error.
Warnings	30	Critical	Some immediate action is required to keep system running well.
	40	Warning	Basic warning.
	50	Minor	Relatively minor warning.
Security	20	Fatal	Operation compromised. For example, no license file exists, no components are licensed, a needed directory cannot be accessed, and similar issues.
	30	Critical	Break-in, connection to Delivers server failed due to access denied, or similar issue.
	40	Security	Access denied to a necessary or requested object
	50	Minor	User logon failed due to invalid password or ID, or similar function.
	55	Trace	Special severity for tracing normal security activity, such as user logged in, user logged out, or similar issue.
Information	20	Fatal	Fatal events that need immediate review.
	30	Critical	Essential start-up, shutdown events.
	40	Informational	Administrative information, such as completion of auto-save.
	45	System	Special severity for important system information, such as cache cleanup started, cache cleanup completed.
	50	Minor	Less important, more granular information, such as session time-out.

Analytics Web Log Filters

FilterRecords customize logging details. Use FilterRecords to specify the implementation (output type) and logging levels for categories of Web logs: Errors, Warnings, Security, and Information.

In the following example, the first two FilterRecords contain the following string:

```
path="saw"
```

This string logs the informational events at level 31, the error messages at level 41, and so on:

```
<FilterRecord writerClassGroup="Cout" path="saw" information="31" warning="41"
  error="41" security="41" />
<FilterRecord writerClassGroup="File" path="saw" information="31" warning="100"
  error="100" security="41" />
<FilterRecord writerClassGroup="File" path="saw.mktgsqlsubsystem.joblog"
  information="41" warning="100" error="100" security="41" />
```

This high-level path applies to every event.

You can customize FilterRecords by adding new FilterRecords, such as the third one shown in the preceding example, with finer-grain specification of log levels for events of various types. In this example, information is being logged to a disk file from saw.mktgsqlsubsystem.log, which generates Marketing job events.

You can turn off logging of job details by changing the information level from 41 to 51, as shown in the following example, or by commenting out the lines:

```
<FilterRecord writerClassGroup="File" path="saw.mktgsqlsubsystem.joblog"
  information="41" warning="100" error="100" security="41" />
<FilterRecord writerClassGroup="File" path="saw.httpserver.request"
  information="51" warning="100" error="100" security="41" />
<FilterRecord writerClassGroup="File" path="saw.httpserver.response"
  information="51" warning="100" error="100" security="41" />
```

The Siebel Analytics Web log filters and their purposes are shown in Table 17 on page 132. (The Analytics Web logging levels are shown in Table 16 on page 130.)

Table 17. Siebel Analytics Web Log Filters

Server Function	Log Filter Name	Log Filter Purpose
Java Host	saw.SAWJavaHost	Siebel Analytics Web Java Host (C++), which runs third-party Java-based applications and communication between those applications and Siebel Analytics Web; for example, for PDF generation.
	saw.sawjavahost.initSAWJavaHost	Runs when the Siebel Analytics Web Java Host is run and initialized.
	saw.sawjavahost.terminateSAWJavaHost	Siebel Analytics Web Java Host shutdown.
Web Service	saw.sawserver	Siebel Analytics Web Service sawserverexe.
	saw.sawserver.initializesawserver	SAWServer initialization.
	saw.sawserver.terminatesawserver	SAWServer shutdown.
	saw.sadis	Siebel Analytics Web Disconnected HTTP Server.
	saw.sadis.receivedata	Request processing by the Analytics Web Disconnected HTTP Server.

Table 17. Siebel Analytics Web Log Filters

Server Function	Log Filter Name	Log Filter Purpose
Web Catalog	saw.catalog.manager.saveCatalogAs	Web Catalog auto save operations.
	saw.catalog.manager.saveCatalog	
	saw.catalog.local.deleteItem	Web Catalog object operations.
	saw.catalog.local.removeFolder	
	saw.catalog.local.moveItem	
	saw.catalog.local.resolveLink	
	saw.catalog.local.createLink	
	saw.catalog.local.readObject	
	saw.catalog.local.writeObject	
	saw.catalog.local.takeOwnership	
	saw.catalog.local.setItemPosition	
	saw.catalog.local.getSubItems	
	saw.catalog.local.createFolder	
	saw.catalog.local.getItemInfo	
	saw.catalog.local.setItemACL	
	saw.catalog.local.getItemACL	
	saw.catalog.replication.error	Web Catalog replication.
	saw.catalog.replication.import.error	Web Catalog Replication import.
	saw.catalog.replication.import	
	saw.catalogPermissionsDialog.setPermissions	Catalog item permissions.
	saw.catalogPermissionsDialog.recurse	
	saw.catalogPermissionsDialog.ui	
Charting	saw.charts.cache	Chart cache.
	saw.config.charts	Chart engine configuration.
	saw.charts.pop	
	saw.charts.pop.formatter	
	saw.charts.pop.embedder.native	
	saw.charts.pop.xslfowriter	
	saw.charts.pop.embedder.http	
	saw.charts.pop.embedder.rpc	

Table 17. Siebel Analytics Web Log Filters

Server Function	Log Filter Name	Log Filter Purpose
RPC	saw.rpc.server.responder	Siebel Analytics Web RPC communication.
	saw.rpc.server	
	saw.rpc.server.socketServer	
	saw.rpc.server.socketServer.cleanup	
Connection Pool	saw.connectionPool	Connection Pool for communication with Analytics Server.
	saw.connectionPool.destructor	
	saw.connectionPool.cleanup	
	saw.connectionPool.getConnection	
Caching	saw.querycache	Query caching.
	saw.querycache.destructor	
	saw.querycache.constructor	
	saw.querycache.cleanup	
	saw.querycache.shutdown	
	saw.querycache.cancelQuery	
	saw.querycache.executeCancel	
	saw.querycache.executeQuery	
	saw.cacheseeding	Cache seeding.
Cube	saw.cube.cache	Cube for pivot table/ crosstab.
	saw.cube.cache.cleanup	
	saw.cube.cache.processCube	
	saw.cube.engine	
	saw.cube.engine.prepareQuery	
	saw.cube.engine.execute	
	saw.cube.engine.execute.dataTraversal	
	saw.cube.engine.execute.calcItems	
	saw.cube.engine.execute.inversion	
HTTP Server	saw.httpserver.request	HTTP requests and responses.
	saw.httpserver.response	
	saw.httpserver.request.soaprequest	SOAP/HTTP requests.
	saw.compressedstream	Compressed stream.
	saw.httpstreamonoutput	HTTP stream.

Table 17. Siebel Analytics Web Log Filters

Server Function	Log Filter Name	Log Filter Purpose
Internationalization	saw.i18n.charsetResolver	Internationalization.
	saw.i18n.charsetResolver.defsLoad	
	saw.i18n.charsetResolver.listsLoad	
	saw.i18n.currencyLookup	
	saw.i18n.currencyLookup.loadCurrencies	
	saw.i18n.localeMapping	
	saw.i18n.localeMapping.load	
	saw.i18n.localeMapping.mapLanguage	
	saw.i18n.localeMapping.mapLocale	
	saw.i18n.xmllocale.applyDefinition	
Marketing List	saw.sqlNodeCacheMgr.loadCatalog	Marketing SQL/list generation and job logging.
	saw.sqlNodeCacheMgr.cleanExpired	
	saw.mktgsqlsubsystem.joblog	
	saw.jobManager.runJob	
	saw.jobManager.cancelJob	
	saw.listGenSubsystem.runListGenerationJob	
	saw.mktgSqlSubsystem.runGetCounts	
	saw.mktgSqlSubsystem.runPurgeJob	
	saw.mktgSqlSubsystem.runPrepareCacheJob	
	saw.mktgSqlSubsystem.runSaveResultSet	
	saw.mktgSqlSubsystem.runDeleteResultSet	

Table 17. Siebel Analytics Web Log Filters

Server Function	Log Filter Name	Log Filter Purpose
ODBC Access	saw.odbc.connection	
	saw.odbc.connection.open	
	saw.odbc.connection.close	
	saw.odbc.connection.construct	
	saw.odbc.connection.connectionString	
	saw.odbc.statement.allocate	
	saw.odbc.statement.bindFields	
	saw.odbc.statement.cancel	
	saw.odbc.statement.close	
	saw.odbc.statement.execute	
	saw.odbc.statement.fetch	
	saw.odbc.statement.execute.sql	
Web Views	saw.views.pivottable.displayer	Pivot table view
	saw.views.pivottable.pdf.displayer	
	saw.views.gfp	Global filter prompt
	saw.views.chart	Chart view
	saw.webviews.compoundview	Compound view
	saw.views.evc	Embedded view controller

Table 17. Siebel Analytics Web Log Filters

Server Function	Log Filter Name	Log Filter Purpose
SOAP-Based Services	saw.SOAP	SOAP dispatch
	saw.SOAP.JobManagementService	SOAP job management
	saw.SOAP.HtmlViewService	HTML retrieval via SOAP
	saw.SOAP.HtmlViewService.addReport	
	saw.SOAP.ReplicationService	SOAP-based replication service
	saw.SOAP.ReplicationService.Export	
	saw.SOAP.ReplicationService.Import	
	saw.SOAP.ReplicationService.PurgeLog	
	saw.SOAP.ReplicationService.MakeForReplication	
	saw.SOAP.ReportEditingService	SOAP-based report editing service
	saw.SOAP.SessionRequestHandler	SOAP session request handler
	saw.SOAP.SoapReportLoader.LoadReport	SOAP catalog object loader
	saw.SOAP.SoapReportLoader.ApplyReportParams	
	saw.SOAP.makeSession	SOAP Session
	saw.SOAP.CatalogService	SOAP-based Catalog service
	saw.SOAP.XMLViewService	SOAP-based XML fetch service
Portal and Dashboards	saw.subsystem.portal	Portal/Dashboard subsystem
	saw.subsystem.portal.pdf	Answers query (PDF generation)
	saw.reportQuery	Answers query object
	saw.fopProxy	Fop proxy (PDF generation)
	saw.delivers.rpc.getDeviceContent.dashboardDelivery	Ibots/Delivers dashboard delivery
	saw.delivers.multipartSerializationProvider	
	saw.pdf.pdfstyle	PDF styles
Filters	saw.subsystem.filters	Filter operations

Table 17. Siebel Analytics Web Log Filters

Server Function	Log Filter Name	Log Filter Purpose
Remote or Disconnected	saw.subsystem.remote.preprocessgenerator	Remote/Disconnected subsystem
	saw.subsystem.remote.generator	
	saw.subsystem.remote.sdcgenerator	
	saw.subsystem.remote.syncSDC	
	saw.subsystem.remote.syncPreProcess	
	saw.subsystem.remote	
Administration	saw.subsystem.admin.upgrade	Admin subsystem Catalog upgrade
Security	saw.subsystem.security	Security subsystem
	saw.subsystem.security.cleanup	
	saw.subsystem.security.hostnameResolver .cleanup	
	saw.subsystem.security.hostnameResolver .processor	
	saw.subsystem.security.hostnameResolver .processor	
System Monitor	saw.systemMonitor	
	saw.systemMonitor.minutely	
	saw.systemMonitor.hourly	
	saw.systemMonitor.daily	
Web Server Startup	saw.webextensionbase.staticinit	
	saw.webextensionbase.init	
	saw.webextensionbase.init.workstationCheck	
Thread Management	saw.threads	
	saw.threadPool	
	saw.threads.syncobjs.conditionwait.wait	
	saw.threads.syncobjs.conditionwait.signal	
Scheduling	saw.taskScheduler	Task scheduler
	saw.taskScheduler.processJob	
	saw.nqscheduler	Scheduler for iBot jobs

Table 17. Siebel Analytics Web Log Filters

Server Function	Log Filter Name	Log Filter Purpose
Miscellaneous	saw.mainwin	Interaction with MainWin (no longer applicable in Siebel Analytics versions 77 and later)
	saw.unknown	Unspecified log source

9 Working with Data and Managing the Web Catalog Using Siebel Analytics Web SOAP API

This chapter 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:

Overview of the Siebel Analytics Web 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. You can also use the Siebel Analytics Web SOAP API to incorporate third-party application content on a Siebel Intelligence Dashboard.

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 SAW 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 chapter.

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 (SAW) WSDL document.

   The following URL is an example URL to access the SAW 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 “Code Example for Writing Web Catalog Information to XML Files Using the Siebel Analytics Web SOAP API” on page 188.

Description of Structures in the Siebel Analytics Web SOAP API

This section 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.

Table 18 lists the structures described in this section and the services that use these structures.

Table 18. Structures and the Services That Use Them

Structures	Services
"AccessControlToken Structure" on page 145	"SecurityService" on page 175
"Account Structure" on page 145	"SecurityService" on page 175
"ACL Structure" on page 145	"SecurityService" on page 175
"CatalogItemsFilter Structure" on page 146	"ReplicationService" on page 167
"CatalogObject Structure" on page 146	"WebCatalogService" on page 177
"Expression Structure" on page 146	All services
"GetSubItemsParams Structure" on page 147	"WebCatalogService" on page 177
"ImportError Structure" on page 147	"ReplicationService" on page 167
"ItemInfo Structure" on page 148	"WebCatalogService" on page 177
"NameValuePair Structure" on page 149	"WebCatalogService" on page 177
"Privilege Structure" on page 149	"SecurityService" on page 175
"ReportHTMLOptions Structure" on page 150	"HtmlViewService" on page 157
"ReportParams Structure" on page 150	All services
"ReportRef Structure" on page 151	All services
"SAColumn Structure" on page 152	"MetadataService" on page 163
"SATable Structure" on page 154	"MetadataService" on page 163
"SASubjectArea Structure" on page 154	"MetadataService" on page 163
"SAWLocale Structure" on page 154	"SAWSessionService" on page 171
"SAWSessionParameters Structure" on page 155	"SAWSessionService" on page 171
"StartPageParams Structure" on page 155	"HtmlViewService" on page 157
"UpdateACLParams Structure" on page 156	"SecurityService" on page 175
"Variable Structure" on page 156	All services

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 19 lists the fields in this structure.

Table 19. 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 20 lists the fields in this structure.

Table 20. 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 21 lists the fields in this structure.

Table 21. 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 22 lists the fields in this structure.

Table 22. 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 23 lists the fields in this structure.

Table 23. 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, see "ItemInfo Structure" on page 148.
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 24 lists the fields in this structure.

Table 24. GetSubItemsParams Structure Fields

Fields	Descriptions
filter	Used for internal purposes only. This 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 25 lists the fields in this structure.

Table 25. ImportError Structure Fields

Fields	Description
String item	The path to the changed item.

Table 25. 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 26 lists the fields in this structure.

Table 26. 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, see "Using Item Signatures in the Siebel Analytics Web SOAP API" on page 142.
NameValuePair[] itemProperties	An array of object properties.

Table 26. 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 27 lists the fields in this structure.

Table 27. 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 28 lists the fields in this structure.

Table 28. 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, see "HtmlViewService" on page 157. Table 29 lists the field in this structure.

Table 29. 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 30 lists the fields in this structure.

Table 30. 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 31 shows how filter expressions are applied to a report.

Table 31. 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 32 shows how variables are applied to a report.

Table 32. 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 33 lists the fields in this structure.

Table 33. 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 34 lists the fields in this structure.

Table 34. 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, see SADataType Values on page 152.
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, see AggregationRule Values on page 153.

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, see *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 35 lists the fields in this structure.

Table 35. 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, see "SAColumn Structure" on page 152.

SASubjectArea Structure

The SASubjectArea structure represents Subject Area attributes. This structure is used in the MetadataService. Table 36 lists the fields in this structure.

Table 36. 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, see "SATable Structure" on page 154.

SAWLocale Structure

The SAWLocale structure defines the locale for the current session. This structure is used in the SAWSessionService. Table 37 lists the fields in this structure.

Table 37. 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 38 lists the fields in this structure.

Table 38. SAWSessionParameters Structure Fields

Fields	Description
SAWLocale locale	The locale to be used, supplied in the SAWLocale structure. For information about the SAWLocale structure, see “SAWLocale Structure” on page 154.
String userAgent	Set this if the HTMLView service will be used with current session. It specifies the userAgent string of the browser, where SAW HTML content is displayed. SAW 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 39 lists the fields in this structure.

Table 39. 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 SAW 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 40 lists the fields in this structure.

Table 40. 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 SAS as well.
int updateFlag	Flag that indicates the mode for UpdateAcl. 0 = Merge the new ACL with the existing one. 1 = Revoke privileges. The new ACL contains a list of accounts and privileges to be revoked. 2 = Replace the existing ACL with the new one.

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 41 lists the fields in this structure.

Table 41. 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.

Description of Siebel Analytics Web SOAP API Methods

This section 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.

For an overview of services and methods, see "Overview of the Siebel Analytics Web SOAP API" on page 141.

This section covers the methods available for the following services:

HtmlViewService

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 42 describes the HTML code excerpts and desired page locations.

Table 42. 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 user sees 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 see 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, see "setBridge() Method" on page 161.

Table 43 shows the supported methods for the HtmlViewService.

Table 43. HtmlViewService Methods

Method Name	Description
addReportToPage() Method on page 158	Adds results to an HTML page.
endPage() Method on page 159	Destroys a server page object and all data associated with it.
getCommonBodyHTML() Method on page 160	Gets HTML to include in the <BODY> section.
getHeadersHTML() Method on page 160	Gets HTML to include in the <HEAD> section.
getHTMLForReport() Method on page 161	Gets HTML to display a particular set of results.
setBridge() Method on page 161	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 162	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, see "startPage() Method" on page 162.
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, see "ReportRef Structure" on page 151.
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, see "Description of Siebel Analytics Web SOAP API Methods" on page 156.
ReportHTMLOptions options	Optional. The display options to apply to the report after execution, supplied in the ReportHTMLOptions structure. For more information, see "ReportHTMLOptions Structure" on page 150.
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, see "startPage() Method" on page 162
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, see "startPage() Method" on page 162.
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, see "startPage() Method" on page 162.
String pageReportID	A character string ID returned by the addReportToPage() method. For information about the addReportToPage method, see "addReportToPage() Method" on page 158.
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, see "How Callback URLs Are Replaced" on page 162.
- 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, see "StartPageParams Structure" on page 155.
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 SAW page ID.

MetadataService

Use the MetadataService to retrieve descriptions of Siebel Analytics schema objects: columns, tables, and subject areas. Table 44 shows the supported methods.

Table 44. MetadataService Methods

Method Names	Description
describeColumn() Method on page 163	Retrieves column information for a specified column in a specified subject area and table.
describeSubjectArea() Method on page 164	Retrieves subject area information for a specified subject area.
describeTable() Method on page 165	Retrieves table information for a specified table in a specified subject area.
getSubjectAreas() Method on page 166	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, see "SAColumn Structure" on page 152.

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, see "SASubjectAreaDetails Values" on page 164.
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 45 lists the available values.

Table 45. SASubjectAreaDetails Values

Values	Description
IncludeTables	Include table list with minimum information about each table.

Table 45. SASubjectAreaDetails Values

Values	Description
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, see "SASubjectArea Structure" on page 154.

Usage

Depending on the value of the detailsLevel parameter, the returned Object contains the information specified in Table 46.

Table 46. 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.

Arguments	Description
SATableDetails detailsLevel	Specifies what information should be retrieved about the table. For information on the SATableDetails structure, see "SATablesDetails Values" on page 166.
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 47 lists the available values.

Table 47. 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, see "SATable Structure" on page 154.

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, see "SASubjectArea Structure" on page 154.

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

Provides methods for use with catalog replication. Table 48 shows the supported methods.

Table 48. ReplicationService Methods

Method Names	Description
export Method on page 167	Exports catalog changes to a specified log file.
import Method on page 168	Import changes from the log file.
markForReplication on page 168	Change the "replicable" flag on a specified folder and its descendants.
purgeLog Method on page 169	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.
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, see "ImportError Structure" on page 147.

markForReplication

Change 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.

Argument	Description
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

Merges arguments and Siebel Analytics Web data to create and return the results. Table 49 shows the supported methods.

Table 49. ReportEditingService Methods

Method Names	Description
applyReportParams() Method on page 169	Applies report arguments to the report object and returns the results.
generateReportSQL() Method on page 170	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, see "ReportRef Structure" on page 151.
ReportParams reportParams	Optional. The filters or variables to apply to the report before execution, supplied in the ReportParams common structure. For more information, see "Description of Siebel Analytics Web SOAP API Methods" on page 156.
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, see "ReportRef Structure" on page 151.
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, see "Description of Siebel Analytics Web SOAP API Methods" on page 156.
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

Defines authentication services such as logon and logoff, and other session-related services. Table 50 shows the supported methods.

Table 50. SAWSessionService Methods

Method Name	Description
getCurUser() Method on page 171	Gets the current user ID for the session.
impersonate() Method on page 172	Logs on and then impersonates the user.
impersonateex() Method on page 172	Logs on and then impersonates the user. Similar to the impersonate method, but impersonateex can specify optional session parameters.
keepAlive() Method on page 173	Instructs Siebel Analytics Web not to end particular sessions due to inactivity.
logoff() Method on page 173	Logs the user off Siebel Analytics Web.
logon() Method on page 173	Authenticates the user.
logonex() Method on page 174	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.
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).

Arguments	Description
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, see "SAWSessionParameters Structure" on page 155.

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, see "Setting the Time to Log Users Off Siebel Analytics Web Automatically" on page 18.

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, see "SAWSessionParameters Structure" on page 155.

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

Provides methods for identifying accounts and privileges. Table 51 shows the supported methods.

Table 51. SecurityService Methods

Method Names	Description
forgetAccount() Method on page 175	Removes a Siebel Analytics Web internal ID to account name mapping.
getGlobalPrivilegeACL() Method on page 175	Get the Access Control List for global privileges.
getGlobalSAWPrivileges() Method on page 176	Get the list of all global privileges.
updateGlobalPrivilegeACL() Method on page 176	Update the Access Control List for global privileges.

forgetAccount() Method

Removes a Siebel Analytics Web internal ID to account name mapping. This 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, see "Account Structure" on page 145.

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, see "ACL Structure" on page 145.

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, see "Privilege Structure" on page 149.

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, see "ACL Structure" on page 145.
UpdateACLParams options	The Access Control List parameters to update, supplied in the UpdateACLParams structure. For information about the UpdateACLParams structure, see "UpdateACLParams Structure" on page 156.
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

Provides methods for navigating and managing the Web Catalog, and reading and writing Web Catalog objects in XML format. Table 52 shows the supported methods.

Table 52. WebCatalogService Methods

Method Names	Description
copyItem() Method on page 177	Copies an object from one location to another in the Web Catalog.
createFolder() Method on page 178	Creates a new folder in the Web Catalog.
createLink() Method on page 178	Creates a link to the Web Catalog.
deleteItem() Method on page 179	Deletes an object from the Web Catalog.
getItemInfo() Method on page 179	Gets Web Catalog information for an object.
getSubItems() Method on page 180	Gets the collection of child subitems for an object in the Web Catalog.
moveItem() Method on page 180	Moves an object in the Web Catalog to a different location in the catalog.
readObject() Method on page 181	Reads an object from the Web Catalog.
removeFolder() Method on page 181	Deletes a folder from the Web Catalog.
setItemProperty() Method on page 182	Sets a property for an object in the Web Catalog.
takeOwnership() Method on page 182	Take ownership on the specified item.
writeObject() Method on page 182	Writes an object to the Web Catalog.
writeReport() Method on page 183	Writes a set of results to the Web Catalog.
writeDashboard() Method on page 184	Writes a dashboard object to the Web Catalog.
writeDashboardPrompt() Method on page 184	Writes a dashboard prompt to the Web Catalog.
writeDashboardPage() Method on page 185	Writes a dashboard page to the Web Catalog.
writeSavedFilter() Method on page 186	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.

Arguments	Description
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, see "removeFolder() Method" on page 181.

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, see "ItemInfo Structure" on page 148.

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, see "GetSubItemsParams Structure" on page 147.
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, see "ItemInfo Structure" on page 148.

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, see "CatalogObject Structure" on page 146.

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, see "deleteItem() Method" on page 179.

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.

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, see "CatalogObject Structure" on page 146. 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, see "CatalogObject Structure" on page 146. 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.

Arguments	Description
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, see "CatalogObject Structure" on page 146. 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.
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, see "CatalogObject Structure" on page 146. 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, see "CatalogObject Structure" on page 146. 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.

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 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, see "CatalogObject Structure" on page 146. 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.

XMLView Service

Retrieves results from Siebel Analytics Web in XML format. Table 53 shows the supported method.

Table 53. XMLView Service Method

Method Name	Description
getResults() Method on page 187	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, see "ReportRef Structure" on page 151.
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, see "ReportParams Structure" on page 150.
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.

Format of Returned Recordsets in the Siebel Analytics Web SOAP API

The following 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.

Code Example for Writing Web Catalog Information to XML Files Using the Siebel Analytics Web SOAP API

This section provides an example of C# code that uses the Siebel Analytics Web SOAP API. The code extracts Web Catalog information and writes 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
   }
}
```

10 Customizing the Siebel Analytics Web User Interface

This chapter describes how to customize the appearance of the Siebel Analytics Web user interface.

NOTE: In Siebel Analytics Web, customization of user interface elements and appearance is accomplished by modifying registry entries, XML message files, and styles and skins, and not through the use of JavaScript. You should not modify JavaScript files located in the folder Web\App\Res in the Siebel Analytics Web installation directory. This is because the objects and methods in these scripts may change, and because these files may be replaced when upgrading. (In a dashboard, users with the appropriate permissions can customize an individual dashboard section by adding HTML to it. This HTML can include JavaScript. For more information, see *Siebel Analytics User Guide*.)

This chapter contains the following topics:

Modifying Siebel Analytics Web User Interface Styles

You can modify the Siebel Cascading Style Sheets (CSS) files and the images stored in the default installation directory to create a custom user interface. The default images and style sheets are located the in the Web\App\Res\s_Siebel7 folder in the Siebel Analytics installation directory. Contained within this directory are of the relevant subdirectories (b_mozilla_4.0, b_nscp, Charts, Images, Maps, Meters, PopBin, Portal, and Views) for the current Siebel style.

Web developers who work with and understand style sheets can modify the default Siebel styles.

NOTE: Modifications to the PopChart appearance files used in charting are not supported by Siebel Systems.

This section contains the following topics:

Working with Cascading Style Sheets to Modify Default Siebel Analytics Web Styles

Web developers who work with and understand style sheets can modify the default Siebel styles. There are three files that affect most of the dashboard user interface:

- **PortalBanner.css.** Influences the overall appearance of a dashboard's top section. This includes the dashboard's names, links, and so on.
- **PortalContent.css.** Influences the overall appearance of a dashboard's main content area.
- **Views.css.** Corresponds to each of the Siebel Analytics request views (Title, Table, Pivot Table, Chart, Narrative, Ticker, and so on).

To see an example default Siebel 7 style sheet

1. Right-click the dashboard and select View Source to view the Web page within a text editor.

   There are several CSS files referenced within the header of the Web page (<HEAD>….</HEAD>).

2. Perform a Find within the document and search for the keyword "class."

   There is a class variable for each available attribute from within one of the CSS files.

   The first class that appears is the PortalBody class (<body class="PortalBody">). If you open up the PortalContent.css file, you see a corresponding section for PortalBody. For example:

   ```
   .PortalBody {
          font-family:  Verdana, Arial, Sans-serif;
          font-size: 9pt;
          background-color: #FFFFFF;
          margin: 0 0 2 0;
   }
   ```

   You could modify the background color of the dashboard page by changing, for example, the hexadecimal color from #FFFFFF (white) to the color of your choice. If you were to save this change, you could go back to the Web browser and click the Refresh button to see the change.

You can change the various CSS classes to adjust the overall look of any Siebel Analytics application. This can be a tedious process, but after you have a good understanding of the available or most commonly used classes, you can perform a find and replace within the text editor to make mass changes to a style.

About Cascading Style Sheet Attributes and Siebel Analytics Web

Cascading style sheets allow Web developers to have control over any object within Siebel Analytics. You can change images, backgrounds, font colors and sizes, table cell gridlines and cell padding, and so on.

For more information about cascading style sheets, you can consult a resource such as the Microsoft Developer Network (MSDN).

Creating a New Dashboard Style for Siebel Analytics Web

The easiest way to create a new style is to copy the s_Siebel7 style folder \\SiebelAnalytics\Web\App\Res\s_Siebel7 and paste it into the data directory at the location \\SiebelAnalyticsData\Web\Res. Copying to the data directory rather than to the main installation directory prevents any customized CSS files and images from being overwritten during a software upgrade.

After the style has been copied, rename the directory from s_Siebel7 to a name that has meaning to you (such as s_ProspectName).

The b_mozilla_4.0 directory contains the important files for making quick changes to the dashboards.

NOTE: The Analytics Web Server service must be restarted before you can see the new style sheet from the Dashboard Properties screen.

To create a new style

1. Copy the s_Siebel7 directory.
2. Paste it to the Res directory in the SiebelAnalyticsData directory and give it a meaningful name.
3. Make and save your modifications.

   The style sheet becomes available in the Dashboard Properties screen when the Analytics Web Server service is restarted.

Example: Customizing the Dashboard Banner Image in Siebel Analytics Web

The image bg_Banner.gif image is referenced on the top section of dashboards. Developers can open the bg_Banner.gif file and make changes, or delete and recreate a new bg_Banner.gif file.

Specifying Defaults for Siebel Analytics Web Styles and Skins

You can specify which style and skin to use when users choose the default style at the Dashboard Properties screen in Siebel Intelligence Dashboards by adding entries to the Siebel Analytics Web configuration file instanceconfig.xml:

- “Specifying Which Siebel Analytics Web Style Folder to Use” on page 196
- “Specifying Which Siebel Analytics Web Skins Folder to Use” on page 196

If users do not make a choice, or if these entries are not present in the file instanceconfig.xml, the Siebel Systems styles and skins are used. These styles and skins are located in the s_Siebel7 and sk_Siebel7 folders in the \Web\App\Res folder in the Siebel Analytics Web installation directory.

For information about working in the configuration file instanceconfig.xml, see “Making Siebel Analytics Web Configuration Changes” on page 11.

Specifying Which Siebel Analytics Web Style Folder to Use

You can specify which style folder to use in the \Web\App\Res folder when users select the Default option from the Styles drop-down list at the Dashboard Properties screen in Siebel Intelligence Dashboards. If your style folder begins with the characters s_, such as s_TestStyle, omit those characters from the entry.

The following entry is an example:

```
<DefaultStyle>TestStyle</DefaultStyle>
```

Specifying Which Siebel Analytics Web Skins Folder to Use

To specify the skins folder that is paired with the style folder you selected as described in "Specifying Which Siebel Analytics Web Style Folder to Use" on page 196, add the following entry. If your skins folder begins with the characters sk_, such as sk_TestSkin, omit those characters from the entry.

The following entry is an example:

```
<DefaultSkin>TestSkin</DefaultSkin>
```

Customizing Siebel Analytics Web NonDashboard Components

Non-dashboard components include Siebel Answers, Siebel Delivers, and Siebel Analytics Web Administration. The CSS files for these components are stored in the sk_ directory within the main installation directory (\\SiebelAnalytics\Web\App\Res\sk_Siebel7). The location b_mozilla_4.0 contains the relevant CSS files that correspond to Siebel Answers, Siebel Delivers, and Siebel Analytics Web Administration, and so on.

Use the same logic that is described in "Modifying Siebel Analytics Web User Interface Styles" on page 193 to make modifications to the non-dashboard components of the Siebel Analytics user interface.

The non-dashboard components are controlled globally. Users cannot toggle between multiple user interfaces for the non-dashboard components.

Customizing the Siebel Analytics Web User Interface Using XML Message Files

This section explains how to customize text elements in message files to manage the default appearance and behavior of the Siebel Analytics Web user interface using XML strings.

NOTE: The intent of this section is to allow organizations that have XML expertise to perform additional customization. If you do not have this expertise, contact Siebel Systems' Services organization, or enlist the assistance of a third party, to help you with customization.

This section contains the following topics:

NOTE: Other topics in this guide describe additional customizations that you can perform by customizing text elements in message files, such as "Configuring Siebel Answers Pivot Table Settings" on page 36.

About the Siebel Analytics Web User Interface XML Message Files

You can customize many of the text elements that appear in Siebel Answers, Siebel Delivers, and on dashboard pages. Examples of text elements include the content of text strings, the text for prompts such as the names of links and buttons, and the text of error and informational messages that are displayed to users as the result of an action.

These text elements are contained in external message files that are distributed with Siebel Analytics Web. The message files are in XML format. Language-specific messages are located in the folder Web\App\Res\l_xx\Messages in the Siebel Analytics Web installation directory, where xx is the language identifier of the selected locale (for example, for english-usa, the identifier is en). Language-independent messages are located in the folder Web\App\Res\Messages in the Siebel Analytics Web installation directory.

You should not edit the message files directly because any changes would not be retained when you install newer versions or service releases. For more information, see "Customizing Siebel Analytics Web XML Messages" on page 198.

How Siebel Analytics Web XML Message Files Are Structured

The name of a particular message file indicates the kind of content that it holds. For example, messages in the file LogonMessages.xml hold message content related to the act of logging on and off the system. Within each XML file, the WebMessage name= tags define the names of the messages. These tags are called *message identifiers*.

A particular message may also reference the content of another message by using a MessageRef tag. For example, the following message in the file LogonMessages.xml references the value of another message:

```
<WebMessage name="kmsgAuthenticateNotLoggedOnToLogOnClickHere">

 <HTML>

  You are not currently logged in to the

  <MessageRef name="kmsgProductServer" />
```

The entry `<MessageRef name="kmsgProductServer" />` in the previous message indicates that the name of the server is taken from the value of the kmsgProductServer message identifier. This message is located in the file ProductMessages.xml, and its value is Siebel Analytic Server:

```
<WebMessage name="kmsgProductServer" CRC="nnnnnnnnnnnnnnnnnnn">

 <TEXT>Siebel Analytic Server</TEXT>
```

Some messages, such as those that contain copyright information and product names, are protected and cannot be changed. If you look in the file ProductMessages.xml, you see text preceding the WebMessage tags indicating that the associated names cannot be changed.

Customizing Siebel Analytics Web XML Messages

This section explains how to change the content of unprotected messages and provides several examples. The intent is to show you how to alter the text of messages, and not to teach you XML.

To customize messages

1. For the messages you want to customize, create like-named message identifiers and customize their text.
2. Create a custom messages folder named CustomMessages.

   NOTE: Organizations that have Siebel Analytics applications may already have a file present in this folder. This file enables Analytics support for Siebel Analytics applications, and should not be modified, moved, or deleted.
3. Place the messages in one or more XML files in the CustomMessages folder, and then place the CustomMessages folder in this location:

   ```
   \SiebelAnalyticsData\Web\Res\l_xx
   ```

   where SiebelAnalyticsData is the name of the data folder created during the installation process, and xx is the language identifier of the selected locale (for example, for english-usa, the identifier is en).

   If you are not concerned with multiple languages, place the folder in the l_en folder. Messages default to l_en if a language-specific version is not found. You need to create the l_xx folder in the analyticsRes folder.
4. Restart Siebel Analytics Server.

You can create as many XML files in the CustomMessages folder as you like, or you can create just one XML file that holds customized messages, for example, CustomMessages.xml.

NOTE: If you plan to support multiple languages, it is recommended that you place control messages, which are messages that are not translated, into one file named CustomControlMessages.xml. Place messages that are translated into another file named, for example, CustomUIMessages.xml. This allows you to place localized versions of the CustomUIMessages.xml file in each language folder as appropriate, such as \SiebelAnalyticsData\Web\analyticsRes\l_de\CustomMessages, \SiebelAnalyticsData\Web\analyticsRes\l_fr\CustomMessages, and so on.

Resolution of Siebel Analytics Web XML Message Name Tags

During initialization, the Analytics Web Server replaces the WebMessage name default text with text from equivalently named tags in any customized XML file, based on the following precedence order, from highest to lowest:

- XML in \SiebelAnalyticsData\Web\Res\l_xx\CustomMessages folder (language- specific folders).
- XML in \SiebelAnalyticsData\Web\Res\l_en\CustomMessages folder (for language-specific user logons if WebMessage name tags reside here, but are not in language-specific files).
- XML in \SiebelAnalyticsData\Web\Res\CustomMessages folder.
- XML in \SiebelAnalytics\Web\App\Res\l_xx\Messages folder.
- XML in \SiebelAnalytics\Web\App\Res\Messages folder.

As an example, when Siebel Analytics Web starts up, it first reads the messages in the folder Web\App\Res\l_xx\Messages in the installation directory, and then reads the messages in the folder \SiebelAnalyticsData\Web\Res\l_xx\CustomMessages. It replaces the default text for those messages with the customized text. If you attempt to alter the text of a protected message, a message is displayed in its place indicating that you attempted this.

Sample Siebel Analytics Web XML Template

The following is a sample template for a CustomMessages.xml file in the folder \SiebelAnalyticsData\Web\analyticsRes\l_xx\CustomMessages. An example CustomMessages.xml file follows the template.

Every message begins with a <WebMessage name=> tag and ends with a </WebMessage> tag. The message text that you can customize is contained between <TEXT> tags or <HTML> tags. To suppress the display, delete the text between the tags.

```
<?xml version="1.0" encoding="utf-8"?>

<WebMessageTables>


    <WebMessageTable system="Custom Messages">

    <!-- The name of a message must match the name of the message you are
overriding. -->

    <WebMessage name="kmsgExampleOverrideMessage">

    <!-- A message can have TEXT and/or HTML versions of it. It is not necessary
to have both. (TEXT will be automatically converted to HTML when necessary). -->


    <TEXT>Example message.</TEXT> <!-- Format used in a text only output -->

    <HTML><b>Example message with bold HTML tags.</b></HTML> <!-- Format used
in an HTML output -->
```

```
    </WebMessage>
</WebMessageTable>
</WebMessageTables>
```

To create a sample template

1. Replicate the sample template in a text editor.
2. Name the file CustomMessages.xml (or any name you choose).
3. Place the file into the CustomMessages folder you have created in the appropriate \SiebelAnalyticsData\Web\analyticsRes\l_xx folder.

Sample Siebel Analytics Web CustomMessages.xml File

The following example shows four customized messages placed in the CustomMessages.xml file.

```
<?xml version="1.0" encoding="utf-8"?>
<WebMessageTables>

<WebMessageTable system="Custom Messages">

<!-- First message -->
    <WebMessage name="kmsgAuthenticateRemembermyIDandpassword">
    <TEXT>Remember my signon name and password.</TEXT>
</WebMessage>

<!-- Second message -->
    <WebMessage name="kkmsgPrivilegeDisplayerAccountUnknown">
    <TEXT>Unknown Account (<Param insert="1"/>). Call the Help Desk at
extension 9999 to set up a new account.</TEXT>
</WebMessage>

<!-- Third message --
    <WebMessage name="kmsgWelcomeFrameCreateNewRequest">
    <HTML>Create a <b>new request</b> by clicking on a Subject Area below. After
creating the request, click on the <b>Done</b> button at the bottom of the page.</
HTML>
```

```
    </WebMessage>

    <!-- Fourth message -->
         <WebMessage name="kmsgUIAdmin">
         <HTML></HTML>
    </WebMessage>

</WebMessageTable>

</WebMessageTables>
```

- The message identifier of the first message being customized is "kmsgAuthenticateRemembermyIDandpassword". The default text for this message is located in the file LogonMessages.xml in the folder Web\App\Res\l_xx\Messages in the installation directory.
- The message identifier of the second message being customized is "kmsgPrivilegeDisplayerAccountUnknown". The default text for this message is located in the file ViewMessages.xml in the folder Web\App\Res\l_xx\Messages in the installation directory. This message contains a variable, (<Param insert="1"/>).

  NOTE: If you are customizing a message that contains a variable, do not alter the variable. In the UNIX environment, be careful to preserve the case of the message name being customized.
- The message identifier for the third message being customized is "kmsgWelcomeFrameCreateNewRequest". The default text for this message is located in the file SearchSysMessages.xml in the folder Web\App\Res\l_xx\Messages in the installation directory. This message is in HTML format and uses an HTML tag (<b>) to display text in bold letters.
- The message identifier for the fourth message being customized is "kmsgUIADMIN". The default text for this message is located in the file UIMessages.xml in the folder Web\App\Res\l_xx\Messages in the installation directory. This message is in HTML format. This message identifier displays the Admin link at the top of each Siebel Answers, Siebel Delivers, or dashboard page. Deleting the Admin text between the <HTML> and </HTML> tags suppresses the display of the link.

Frequently Customized Siebel Analytics Web User Interface Messages

Messages that are frequently customized are located in the file UIMessages.xml. This file contain text strings for display elements and for links throughout the user interface.

For example, the following messages contain the text for the links Admin, Alerts!, and Answers in the user interface. You can customize the text, or delete the text to suppress the display of the link.

```
<WebMessage name="kmsgUIAdmin">
  <HTML>Admin</HTML>
</WebMessage>

<WebMessage name="kmsgUIAlerts">
  <HTML>Alerts!</HTML>
</WebMessage>

<WebMessage name="kmsgUIAnswers">
  <HTML>Answers</HTML>
</WebMessage>
```

For example, users can export the data for a request to a Microsoft Excel file. If you do not want to support the download to Excel option, you can remove the link.

To remove the Download link

1. Open the ViewsControlMessages.xml file.
2. Copy the following message to the CustomMessages.xml file:

```
<WebMessage name="kmsgEVCDownloadLinks">
  <HTML>
    <a insert="1">
      <MessageRef name="kmsgEVCLinkDownloadExcel" />
    </a>

    <a insert="2">
      <MessageRef name="kmsgEVCLinkDownloadData" />
```

```
    </a>
  </HTML>
```

3 Delete the first message identifier anchor:

```
<a insert="1">
  <MessageRef name="kmsgEVCLinkDownloadExcel" />
</a>
```

4 Save the CustomMessages.xml file.

For more information about the CustomMessages.xml file, see "Sample Siebel Analytics Web CustomMessages.xml File" on page 200.

Customizing the Appearance of the Siebel Analytics Web Logon Screens

Users must log on to Siebel Analytics Web to gain access to Siebel Intelligence Dashboards, Siebel Answers, and other Siebel Analytics Web components. The logon process provides a user interface to the Siebel Analytics Server authentication process.

When Siebel Analytics Web is accessed, the user is presented with the default Siebel Analytics Web logon screen. The user must enter an appropriate Siebel Analytics user name and password. When authentication is complete, the user gains access to the appropriate Siebel Analytics Web components, and the user's default dashboard displays. For more information, see "About Siebel Analytics Web User Authentication" on page 102.

To customize the appearance of the logon screens

- Override the relevant messages in the file LogonControlMessages.xml.

  NOTE: The Not Logged On screen (kmsgAuthenticateNotLoggedOn), used for session timeout, appears only when users are not logged on and attempt to access a URL that does not support direct logon. For example, suppose a user accesses Siebel Answers and clicks the Log Off link. If the user clicks the browser's Back button and then clicks the My Account link, the user receives the Not Logged On screen.

For general information about customization the user interface with XML messages, see "Customizing the Siebel Analytics Web User Interface Using XML Message Files" on page 196.

NOTE: This feature does not allow the Siebel Analytics Web administrator to control how users are actually authenticated. For information about authentication options, see "About Siebel Analytics Web User Authentication" on page 102.

Configuring the Siebel Analytics Web ReportUI Portlet

The Siebel Analytics Web ReportUI portlet is a software component that displays Siebel Analytics Web content in a portlet in WebSphere Portal. The Siebel Analytics Web ReportUI portlet adheres to WebSphere portlet specifications. Its implementation is based on an existing HTML view service provided by the Siebel Analytics Web SOAP layer.

The connection schema works as follows:

1. The browser displaying the WebSphere portal contains the Siebel Analytics Web ReportUI portlet which requests a Siebel Analytics Web report.
2. The WebSphere Portal Server receives the request and forwards it to the Bridge servlet in the Siebel Analytics Web Portlets Web Application.
3. The Bridge servlet contacts the Siebel Analytics Web server.
4. The Siebel Analytics Web server delivers the report information to the ReportUI portlet in the Siebel Analytics Web Portlets Web Application.
5. The ReportUI portlet sends the information to the WebSphere Portal Server which forwards it on to the user's web browser for display.

Installing and Configuring the Siebel Analytics Web ReportUI Portlet

These instruction assume you have installed the WebSphere Portal Server according to the manufacturer's instructions and have started the WebSphere Portal Server. A complete installation consists of the following activities:

- "Installing the Siebel Analytics Web ReportUI Portlet" on page 204
- "Creating a Credential Vault" on page 205
- "Editing the WebSphere Page Layout" on page 206
- "Adding the Web Application Parameters" on page 207

Installing the Siebel Analytics Web ReportUI Portlet

The Siebel Analytics Web ReportUI portlet is delivered in the form of a Web Archive (WAR) file. The WebSphere Portal reads in this WAR file and installs the portlet.

To install the Siebel Analytics Web ReportUI portlet

1. In the IBM WebSphere Portal Welcome page, click Administration.
2. Click Portlet Management, then click Web Modules, and then click Install.

3 In the Directory field, enter the full path and file name to the sawwsportlets.war file.

The default location is
<Analytics Root>\Web\SDK\Portlets\Websphere\sawwsportlets.war

4 Click Next.

The Manage Web Modules screen displays the name of the portlet to be installed.

5 Click Finish.

WebSphere uploads and installs the Siebel Analytics Web ReportUI portlet. If it installs correctly, the Manage Web Modules screen displays the following message.

EJPAQ1332I: Web module was successfully installed.

Creating a Credential Vault

You need to create a credential vault for storing the Siebel Analytics Web administrator user name and password. The portlet uses these to authenticate WebSphere users against Siebel Analytics Web through the process of impersonation, in the same way that Siebel Delivers logs on to Siebel Analytics Web as the administrator and then impersonates the user who wants an iBot run. Thus, the WebSphere user, must also exist in the Siebel Analytics Repository. WebSphere logs into Siebel Analytics Web using the administrator's credentials but tells Siebel Analytics Web that the user logging in is the WebSphere user.

To create a credential vault

1 In the IBM WebSphere Portal Welcome screen, click Administration.

2 Click Access, then click Credential Vault.

3 In the Credential Vault screen, click the Add a vault slot button.

4 In the Select Vault screen, fill in the fields using the following values, then click OK.

Field	Value
Vault	Default
Name	sawseveradmin
Vault Segment	DefaultAdminSegment
Vault slot is shared	Checked
Shared userid	Enter a user ID that has Administrative privileges.
Shared password	Enter the password for the account.

5 Click Portlet Management, then click Applications.

6 In the Manage Applications screen, use the search function to locate the Siebel Analytics Web Portlets default concreate application.

Suggestion: Search by “Title starts with” and search for the term “Siebel Analytics”.

7 In the Portlet applications list click the pencil icon to the right of the Siebel Analytics Web Portlets default concreate application row.

NOTE: WebSphere’s Manage Application’s screen cannot accept modifications to existing parameters. You must delete any existing parameters and add new parameters with the new values.

8 In the Manage Applications screen, delete any existing parameters and enter the following parameters and click OK.

Parameter	Value
com.siebel.analytics.web.portlets.websphere.SuperuserSlotId	The name of WebSphere Credential Vault Slot which holds the administrative user name and password for the Siebel Analytics Web server identified by the URL above. This is the name you entered in Step 4 on page 205.
com.siebel.analytics.web.SAWServer.URL	The access URL of the Siebel Analytics Web server. It includes the part of the URL before the question mark. It usually ends with saw.dll. Default value is: `http://localhost/analytics/saw.dll`

Editing the WebSphere Page Layout

After you have installed the Siebel Analytics Web ReportUI portlet, you need to add it to the WebSphere Portal page.

To edit the WebSphere page layout

1 In the IBM WebSphere Portal Welcome page, click Edit Page.

2 In the Edit Layout screen, click Add portlets in one of the available content columns.

3 Use the search function to locate the Siebel Analytics Web ReportUI Portlet, and then click OK.

   Suggestion: Search by “Title starts with” and search for the term “Siebel Analytics”.

   The Edit Layout screen redisplays, showing the Siebel Analytics Web ReportUI Portlet in the selected content column.

4 Click Done.

   The WebSphere Portal Welcome screen appears. In the portlet location where you added the Siebel Analytics Web ReportUI Portlet an error message appears indicating that the report path is not specified. This is expected.

5 Click the pencil icon to the right of the Siebel Analytics Web ReportUI Portlet.

6 In the Configure Siebel Analytics Web ReportUI Portlet screen, enter the report path of the report to appear in the portlet, and then click Save.

Make sure that the SOAP API access is licensed on the Siebel Analytics Web server. If it is not, then no reports can be returned to the WebSphere Portlet. Also, the username of the user who logs into WebSphere must have access to the report you are trying to display in the portal.

Adding the Web Application Parameters

Set the following parameters in the `web.xml` file for the sawwsportlets web application. Add the `<context-param>` child node to the `<web-app>` element like this:

```
<context-param>
   <param-name>com.siebel.analytic.web.EncryptionPassword</param-name>
    <param-value>password</param-value>
</context-param>
```

Where *password* is a key used to encrypt the Siebel Analytics Web server URL while passing it from portlet to bridge servlet. If this parameter is not set a random password is generated.

11 Integrating Siebel Analytics Web into Corporate Environments Using HTTP

This chapter explains the HTTP methods that you can use to integrate Siebel Analytics Web into your corporate environment. Siebel Analytics Web provides entry points for most functionality exposed in the system. Calls can be made through HTTP using a simple URL syntax or JavaScript commands. In UNIX, interfaces are exposed through a Java Servlet. This chapter contains the following topics:

Incorporating Siebel Analytics Results into External Portals or Applications Using the Go URL

This section describes how to use the Siebel Analytics Web Go URL to incorporate results into external portals or applications. It contains the following topics:

About the Siebel Analytics Web GO URL

The Siebel Analytics Web Go URL command is for use in incorporating specific Siebel Analytics results into external portals or applications. The Go URL is used when you add a result to your favorites, or add a link to a request to your dashboard or an external Web site. It has a number of forms and optional arguments that can be used to control its behavior.

You can post the Go URL as a Form or issue it as a URL. If you are issuing parameters as part of a URL, they need to be escaped properly. You need to replace spaces with plus (+) signs, and so on. For example, to pass East Region as a value, type East+Region.

When called from within a Siebel Analytics Web screen, such as a dashboard or an HTML result view, the URL should begin with the following characters:

```
saw.dll?Go
```

When called from another screen on the same Web server, the URL should begin with the following characters:

```
/Analytics/saw.dll?Go
```

When referenced from a screen on a different server (or sent through email, and so on), the URL should begin with the fully qualified server name or IP address:

```
http://server_name_or_ip_address/Analytics/saw.dll?Go
```

To test these commands, you can enter the fully qualified version into the Address field in Internet Explorer.

Structure of the Basic Siebel Analytics Web Go URL

The basic Go URL command needs the full Web Catalog path to the request you want to execute. It returns the default result view, as defined in the request.

For example, the following go URL command returns the default result view as defined in the request, where SB2 is the name of the request to execute.

```
saw.dll?Go&Path=/Shared/Test/SB2
```

Optional Parameters for the Siebel Analytics Web Go URL

You can modify the behavior of the Go URL command by adding one or more of the following parameters. If an invalid URL is specified (for example, you type a parameter incorrectly), the browser displays a “The page cannot be found” error with the detailed text of “HTTP 400 - Bad Request.”

NOTE: In parameter descriptions, SB2 is the name of the request to execute.

- **User ID and Password.** If omitted, the user is prompted for user ID and password information, unless the user chose the option to have logon information remembered when last logged on.

  This is the format, where uuu is the user ID and ppp is the password:

  ```
  &NQUser=uuu&NQPassword=ppp
  ```

  Example:

  ```
  saw.dll?Go&Path=/Shared/Test/SB2&NQUser=user1&NQPassword=rock
  ```

  This logs on as user1 with a password of rock, and executes the request.

- **Link Options.** The results will include links.

  This is the format:

```
&Options=x
```

The x can be one or more of the following letters:

Letter	Link
m	Modify Request
f	Printer Friendly
d	Download to Excel
r	Refresh Results

Example:

```
saw.dll?Go&Path=/Shared/Test/SB2&Options=md
```

This displays results with the links Modify Request and Download.

- **Printer Friendly.** Results are in a printer-friendly format, without the paging controls, hot links, and so on.

  This is the format:

  ```
  &Action=print
  ```

  Example:

  ```
  saw.dll?Go&Path=/Shared/Test/SB2&Action=Print
  ```

- **Application Friendly.** Results are displayed in an application-friendly format, such as for Microsoft Excel, without the paging control, hot links, the Powered by Siebel Analytics phrase, and so on.

  This is the format:

  ```
  =&Action=Extract
  ```

  Example:

  ```
  saw.dll?Go&Path=/Shared/Test/SB2&Action=Extract
  ```

  The Extract action also acts as a Navigate action (see "Passing Filters to the Siebel Analytics Web Go URL Through a URL (Navigation)" on page 214) so you can filter the results that are returned by the call.

- **Specific View.** This shows an individual result view rather than the default compound view.

  This is the format, where xx is the name of the view:

  ```
  saw.dll?Go&Path=/Shared/Test/SB2&ViewName=xx
  ```

  Example:

  ```
  saw.dll?Go&Path=/Shared/Test/SB2&ViewName=Chart
  ```

  Assuming that the request contains a Chart view named Chart, this displays just the Chart view.

- **Specific Style.** This shows the results using a specified style. If the style does not exist, the default is used.

  This is the format, where xx is the name of the style:

  ```
  saw.dll?Go&Path=/Shared/Test/SB2&Style=xx
  ```

  Example:

  ```
  saw.dll?Go&Path=/Shared/Test/SB2&Style=Lime
  ```

  This uses the style named Lime to show the results.

- **Result Format.** This controls the format of the results.

  This is the format, where xx is XML or HTML:

  ```
  saw.dll?Go&Path=/Shared/Test/SB2&Format=xx
  ```

  Example:

  ```
  saw.dll?Go&Path=/Shared/Test/SB2&Format=XML
  ```

  This shows results in XML.

Displaying All Records in a Table

There are two ways to display all the records in the table:

- Set the Rows per Page property on the Table view to 10,000, and then use the basic Go. This is the easier of the two methods.
- Issue the following URL, assuming the View you want to control is called Table:

  ```
  saw.dll?Go&Path=/Shared/Test/SB2&Action=Scroll&P5=-1&ViewID=go~Table
  ```

Referencing Dashboard Content in External Portals or Applications Using the Siebel Analytics Web Dashboard URL

This section describes how to use the Siebel Analytics Web Dashboard URL. It contains the following topics:

About the Siebel Analytics Web Dashboard URL

The Siebel Analytics Web Dashboard URL is for use in incorporating or referencing the content of a specific Siebel Intelligence Dashboard in external portals or applications. It has a number of forms and optional arguments that can be used to control its behavior.

You can post the Dashboard URL command as a Form or issue it as a URL. If you are issuing parameters as part of a URL, they need to be escaped properly. You need to replace spaces with plus (+) signs, and so on. For example, to pass East Region as a value, type East+Region.

When called from within a Siebel Analytics Web screen, such as a dashboard or an HTML result view, the URL should begin with this:

```
saw.dll?Dashboard
```

When called from another screen on the same Web server, the URL should begin with this:

```
/Analytics/saw.dll?Dashboard
```

When referenced from a screen on a different server (or sent through email, and so on), the URL should begin with the fully qualified server name or IP address:

```
http://server_name_or_ip_address/Analytics/saw.dll?Dashboard
```

To test these commands, you can enter the fully qualified version into the Address field in Internet Explorer.

Structure of the Basic Siebel Analytics Web Dashboard URL

The basic Dashboard URL command needs no parameters. It displays the user's default portal, after authenticating the user.

This is the format:

```
saw.dll?Dashboard
```

Optional Parameters for the Siebel Analytics Web Dashboard URL

You can modify the behavior of the dashboard URL command by adding one or more of the following parameters. If an invalid URL is specified (for example, you type a parameter incorrectly), the browser returns a "The page cannot be found" error with detailed text of "HTTP 400 - Bad Request."

NOTE: In parameter descriptions, SB2 is the name of the request to execute.

User ID and Password. If omitted, the user is prompted for user ID and password information, unless the user chose the option to have logon information remembered when last logged on. If using a Session ID or Ticket, pass it as the NQUser parameter.

This is the format, where uuu is the user ID, and ppp is the password:

```
&NQUser=uuu&NQPassword=ppp
```

Example:

```
saw.dll?Go&Path=/Shared/Test/SB2&NQUser=user1&NQPassword=rock
```

This logs on as user1 with a password of rock, and displays the dashboard.

PortalPath. This parameter allows you to direct the user to a specific dashboard. If the user does not have permission to the dashboard, an Access Denied error is returned.

This is the format, where ppp is the fully qualified path of the dashboard in the Web Catalog, and x+x is the name of the dashboard to display.

```
saw.dll?Dashboard&PortalPath=/Shared/Test/_Portal/x+x&NQUser=sessionid
```

Example:

```
saw.dll?Dashboard&PortalPath=/Shared/Test/_Portal/Sales+Dashboard&NQUser=sessionid
```

This authenticates the user using the provided session ID, and displays the Sales Dashboard.

Using the Siebel Analytics Web Go URL to Issue SQL and Pass Filters

This section explains how to use the Go URL command to issue SQL, and how to pass filters to be used for navigation. It contains the following topics:

Issuing SQL Using the Siebel Analytics Web Go URL

The Go URL command can be used to issue Siebel Analytics SQL. These forms of the Go URL return tabular results. The basic options from &Style= and &Options= can be used here as well.

To issue Siebel Analytics's simplified SQL, include the escaped SQL as a parameter to the Go URL. For example:

```
saw.dll?Go&SQL=select+Region,Dollars+from+SupplierSales
```

where the FROM clause is the name of the Subject Area to query.

Alternatively, the command IssueRawSQL can be used to bypass the Web processing and issue SQL directly against the Analytics Server.

Passing Filters to the Siebel Analytics Web Go URL Through a URL (Navigation)

The Go URL can also be used to pass context such as filters to a destination request. This is done by adding additional parameters to the call. You need to make sure that any columns you are passing are set up in the destination with Is Prompted filters, or specific default filters.

Navigation Parameters

The basic syntax of the navigation command is the same as presented in the section "Structure of the Basic Siebel Analytics Web Dashboard URL" on page 213, but with the addition of the Action=Navigate parameter, and then population of the P1 - Pn parameters, as necessary.

`&Action=Navigate`

`&P0=n` where n is the number of columns you wish to filter, currently 1 - 6.

`&P1=op` where op is one of the following operators.

Operator	Meaning
eq	Equal to or in.
neq	Not equal to or not in.
lt	Less than.
gt	Greater than.
ge	Greater than or equal to.
le	Less than or equal to.
bwith	Begins with.
ewith	Ends with.
cany	Contains any (of the values in &P3).
call	Contains all (of the values in &P3).
like	You need to type %25 in place of the usual % wildcard. See the examples that follow.
top	&P3 should contain 1+n, where n is the number of top items you want to see.
bottom	&P3 should contain 1+n, where n is the number of bottom items you want to see.
bet	Between (&P3 must have two values).
null	Is null (&P3 must be 0 or omitted).
nnul	Is not null (&P3 must be 0 or omitted).
&P2=ttt.ccc	In this parameter, ttt is the table name and ccc is the column name. If the table or column contains spaces, it must be quoted with double-quotes. Spaces should be escaped as %20, for example, Measures."Dollar%20Sales".
&P3=n+xxx+yyy+...+zzz	In this parameter, n is the number of values, and xxx, yyy, and zzz are the actual values. **NOTE:** If the value of P3 begins with a numeric character, the entire value must be enclosed in quotes. For example: `saw.dll?Go&Path=/Shared/Test/SB2&Action=Navigate&P0=1&P1=top&P2=Customers.Region&P3="7West"`

NOTE: The settings for &P1,&P2, and &P3 are repeated for &P4-P6, &P7-P9, &P8-P10, &P11-P13,

&P14-P16, and &P17-P19 as necessary, depending on the value of &P0.

Navigation Examples

This returns records for the East and Central regions:

```
Saw.dll?Go&Path=/Shared/Test/
SB2&Action=Navigate&P0=1&P1=eq&P2=Customers.Region&P3=2+Central+East
```

This returns records for like Regions E....t:

```
saw.dll?Go&Path=/Shared/Test/
SB2&Action=Navigate&P0=1&P1=like&P2=Customers.Region&P3=1+E%25t
```

This returns the top two regions by dollars sold:

```
saw.dll?Go&Path=/Shared/Test/
SB2&Action=Navigate&P0=1&P1=top&P2="Sales%20Facts".Dollars&P3=1+2
```

This is an example where the number of arguments is not included in the syntax:

```
saw.dll?Go&Path=/Shared/Test/
SB2&Action=Navigate&P0=1&P1=top&P2=Customers.Region&P3=Central
```

NOTE: You can omit the number of arguments only if just one argument value is included.

This returns records with between 2,000,000 and 2,500,000 in sales:

```
saw.dll?Go&Path=/Shared/Test/
SB2&Action=Navigate&P0=1&P1=top&P2="Sales%20Facts".Dollars&P3=2+2000000+2500000
```

This returns records for Regions beginning with the letter E:

```
saw.dll?Go&Path=vate&P0=1&P1=bwith&P2=Customers.Region&P3=1+E
```

This returns records for Regions containing the letter E and having more than 20 million in sales:

```
saw.dll?Go&Path=/Shared/Test/
SB2&Action=Navigate&P0=2&P1=cany&P2=Customers.Region&P3=1+e&P4=gt&P5="Sales%20Fact
s".Dollars&P6=1+20000000
```

Siebel Analytics Web navigation is currently supported from charts, table and pivot table views, HTML views, and external applications and Web pages. The destination search should have filters defined on columns for which it wants to receive context. These can be specific filters or, usually, the Is Prompted filter. In addition to the Table.Column value specifically referenced in the navigation call, all filters from the source request that have corresponding table.columns in the destination, are applied to the destination. Therefore, the appropriate context from a source can be passed to the destination.

Navigation from Charts

From the chart properties screen, check the Navigate radio button and enter the full path to the saved search or portal that you are interested in the field provided, for example:

```
/shared/topaz/performance/transaction details
```

Navigation Using JavaScript

Navigation can currently be accomplished using the custom text/date formatter for a column. The central concept is that you add a column you want to navigate from to your search. You then choose Custom Text Format from the properties for the column, and enter HTML that calls one of the two provided JavaScript functions. This technique can be used to perform many actions, including sorting columns, calling custom JavaScript functions, and so on.

Siebel Analytics Web includes two JavaScript functions that enable navigation from Table and Pivot views: GoNav and PortalNav. (These functions are located in /res/mozilla/viewhelper.js if you are interested in seeing what they do.) The former handles navigation to a specific search. The latter handles navigation to a specific dashboard. A description of their syntax follows, along with example Custom Text formats that you can use to accomplish navigation.

NOTE: Siebel Analytics Web is standardized on the class=Nav, in order to control the look of the navigable text using the style sheet.

`function GoNav(sPath,sTbl,sCol,sVal,sTarget)` where

`sPath` = the catalog path of the destination search.

`sTbl` = the logical table name to filter.

`sCol` = the logical column name to filter.

`sVal` = the value to filter by.

`sTarget` (optional) = "_blank" to open a new browser window with the results.

Sample Custom Text Format for GoNav Call (include the quotes):

```
"<font class=nav onclick=\"JavaScript:GoNav('/shared/topaz/performance/transaction
details','Transaction','Quality','"@"');\">"@"</font>";;@
```

`function PortalNav(sPortal,sTbl,sCol,sVal)`

`sPortal` = the catalog path of the destination portal.

`sTbl` = the logical table name to filter.

`sCol` = the logical column name to filter.

`sVal` = the value to filter by.

Sample Custom Text Format for PortalNav Call (include the quotes):

```
"<font class=nav onclick=\"JavaScript:PortalNav('/shared/topaz/_portal/transaction
analysis','Transaction','Type','"@"');\">"@"</font>"
```

Navigation from HTML Results

This is the same as described in "Navigation Using JavaScript" on page 217, but rather than using a custom formatter, type in the HTML syntax with static values in place of the @ signs.

Example of a Siebel Analytics Third-Party SQL Tool Integration

This section illustrates the requirements for integrating a third-party SQL tool with Siebel Analytics by describing an example integration, using Microsoft Access. Because Siebel Analytics is designed as a middleware platform for enterprise data access and integration, common report writers and business intelligence tools can communicate natively with the Siebel Analytics Server.

Most third-party SQL tools require the user to include join conditions within queries, in order to avoid cross-joins. A cross-join occurs when a request does not have a WHERE clause, which, in turn creates a Cartesian product of the tables involved in the join. The size of a Cartesian product is the number of rows in the first table multiplied by the number of rows in the second table.

To integrate Microsoft Access with the Siebel Analytics Server, the Siebel Analytics Server administrator must expose the keys within the Presentation layer of the Siebel Analytics Administration Tool.

Example of integrating a third-party SQL tool

1. Drag and drop the keys from the Business Model and Mapping layer to the Presentation layer and save the repository.
2. Open Microsoft Access, select the option Blank Access Database, type the name siebel-analytics.mdb when prompted, and click the Create button.
3. After creating the new Microsoft Access database, right-click in the white section of the screen and select Link Tables.
4. From the Files of Type drop-down list box, select ODBC Databases.

   The Select a Source Dialog appears, and prompts you for a Data Source Name.
5. Click the Machine Data Source tab, locate the Analytics_Web DNS, and click OK.

   The Siebel Analytics Server requires a login.
6. Type your user ID and password.

   The Import Objects dialog box appears.
7. Click the Select All button, or highlight the desired logical tables from Siebel Analytics.

   The import may take a while to complete.
8. When the import completes, right-click in the white section of the screen and select Relationships:
   a. Add the desired tables and drag and drop the keys from the dimension tables (Period, Market, Product) to the fact table (Sales Measures).
   b. Drag and drop Period Key over perkey, and repeat for each corresponding key, to create the joins.

   Now, you can test and run a request.
9. Select Create query in Design view from the Queries button:

a Select Markets, Products and Sales Facts.

b Add Region, Brand, Units and Dollars, respectively, then click the Run button.

Index

B

C

D

J

L

M

N

O

P

Q

R

S

T

U

V

W

X