SIEBEL 7
eBusiness

SIEBEL ANALYTICS SCHEDULER GUIDE

VERSION 7.7
DECEMBER 2003

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404

Contents

Chapter 1. Introduction

Chapter 2. Using Siebel Analytics Scheduler

Index

Introduction 1

Siebel Analytics Scheduler Guide provides information about Siebel Analytics Scheduler, a component of Siebel Analytics that you use to schedule jobs and reports.

Intended users of this guide include administrators for the Siebel Analytics environment, architects of the decision support applications created with Siebel Analytics software, and managers who are responsible for the installation, development, and system administration of decision support applications.

This book is primarily useful to people whose titles or job descriptions match one of the following:

Database Administrators	Persons who administer the database, including data loading, monitoring, backup and recovery, space allocation and sizing, and user account management.
Siebel Analytics Architects	Architects of the decision support applications created with Siebel Analytics software.
Siebel Application Administrators	Persons responsible for planning, setting up, and maintaining Siebel applications.
Siebel Application Developers	Persons who plan, implement, and configure Siebel applications, possibly adding new functionality.
Siebel System Administrators	Persons responsible for the whole application environment, including installing, maintaining, and upgrading Siebel applications.

This guide assumes that you are knowledgeable in the areas of relational databases, decision support applications, dimensional design, and the operating system(s) under which you are running the Siebel Analytics components.

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 Bookshelf. To find out more about the modules your organization has purchased, see your corporate purchasing agent or your Siebel sales representative.

What's New in Siebel Analytics Scheduler, Version 7.7

Table 1 lists changes described in this version of the documentation to support version 7.7 of the software.

Table 1. New Product Features in Siebel Analytics Scheduler, Version 7.7

Topic	Description
Background process. See "Scheduler as a Background Process" on page 21.	Information is provided on how to run Siebel Analytics Scheduler as a background process.
Job Manager configuration. See "Job Manager Configuration—Scheduler Tab" on page 15, "Job Manager Configuration—Siebel Mail Tab" on page 18, "Job Manager Configuration—Siebel iBots Tab" on page 19, and "Job Manager Configuration—Workflow Tab" on page 20.	The Job Manager has a new tab and new fields on existing tabs.
Siebel Delivers can trigger Siebel eBusiness Applications workflows. See "Integrating Workflows with Siebel Delivers" on page 34 and "Job Manager Configuration—Workflow Tab" on page 20.	Siebel Delivers supports native integration with Siebel Workflow 7.5 so that iBots can trigger workflow tasks.

Table 1. New Product Features in Siebel Analytics Scheduler, Version 7.7

Topic	Description
The NQSConnect method has been deprecated.	Removed the topic "NQSConnect Method" from the documentation. For other options, contact a support representative.
The Siebel Object has been deprecated.	Removed the topic "Siebel Object Methods" from the documentation. For other options, contact a support representative.

Related Documentation

For more information on Siebel Analytics, read the following books.

- *Siebel Analytics Installation and Configuration Guide*
- *Siebel Analytics Server Administration Guide*
- *Siebel Analytics Message Reference*
- *Siebel Analytics User Guide*
- *Siebel Analytics Web Administration Guide*

Using Siebel Analytics Scheduler 2

Siebel Analytics Scheduler is an extensible scheduling application for scheduling reports to be delivered to users at specified times. It is the engine behind the Siebel iBots feature of Siebel Delivers, and is used by the Job Manager feature of Siebel Analytics Administration Tool. In Windows, Siebel Analytics Scheduler runs as a Windows service. In Sun Solaris, IBM AIX, and Hewlett-Packard HP-UX UNIX environments, it runs as a process.

Concepts and Terms Related to Analytics Scheduler

Siebel Analytics Scheduler manages and schedules jobs. A job is a task performed by Siebel Analytics Server. The Scheduler supports two types of jobs: scripted jobs that you set up and submit using the Job Manager feature of Siebel Analytics Administration Tool; and unscripted jobs, called iBots, that you set up and submit using Siebel Delivers.

For scripted jobs, Siebel Analytics Scheduler supports two scripting languages: VBScript and Jscript. An example of a scripted job would be to take the Siebel Analytics Server usage statistics that are logged in a file and periodically load them into a back-end database. The script would define the actions to be performed and when the actions should be executed.

To create an iBot, you define the actions to be performed using Siebel Delivers. Siebel Analytics Web gathers the necessary information about the priority, delivery devices, user, and other characteristics; packages that information into a job; and tells the Scheduler when it wants the job to be executed.

NOTE: To create iBots, you must have licensed Siebel Delivers. If you have not licensed Siebel Delivers, you can only create scripted jobs using the Job Manager feature of Siebel Analytics Administration Tool.

If you are setting up the Scheduler to use only iBots, you do not have to set up separate jobs using the Job Manager interface. However, because iBots are jobs, you can create iBots using Siebel Delivers and then further configure the iBots using the Job Manager. To do so, you must first configure the Scheduler. For information about configuration options, see "Configuring Siebel Analytics Scheduler" on page 14.

For information about setting up iBots and making them available for subscription, see the online help for Siebel Delivers. For information about setting up and managing scripted jobs, see "Setting Up Analytics Scheduler Jobs" on page 24, "Working with Scheduler Job Instances" on page 33, and "Working with Scheduler Job Scripts" on page 36.

The Back-End Database for Analytics Scheduler

Siebel Analytics Scheduler uses a single commercial back-end database to store pertinent information about a job, its instances, and its parameters.

The Scheduler service cannot start if it cannot access the back end database. Make sure this database is configured and operational. For information about the back-end databases supported by Siebel Analytics Scheduler, see the system requirements and supported platforms documentation for your Siebel application.

Schemas

The associated schemas are located in the Schema directory in the Siebel Analytics software installation directory. This directory holds several SQL scripts that can be used by major commercial databases to create the tables in the Scheduler's schema.

The schemas include four tables: S_NQ_JOB, S_NQ_INSTANCE, S_NQ_JOB_PARAM, and S_NQ_ERR_MSG. The contents of these tables are implementation specific. For information about creating these tables, see *Siebel Analytics Installation and Configuration Guide*.

NOTE: The Siebel Analytics Scheduler tables are included in the Siebel eBusiness Applications OLTP, version 7.5 and later. If you have version 7.5 (or later) of Siebel eBusiness Applications installed, you do not need to install the Siebel Analytics Scheduler tables. The Siebel Analytics Scheduler tables are not included in versions of Siebel eBusiness Applications prior to Version 7.5. If you are upgrading from any version prior to Version 7.5, contact Technical Support for assistance.

Table 2 describes the information stored in each table.

Table 2. Schema Tables

Table	Description
S_NQ_JOB	Stores information about scheduled jobs.
S_NQ_INSTANCE	Stores information about instances.
S_NQ_JOB_PARAM	Stores information about job parameters.
S_NQ_ERR_MSG	Stores information about job instances that do not complete successfully.

Do not change the table or column names; they are used internally by Siebel Analytics Scheduler. The data types for each column should remain true to the intent of the schema. For example, if the job ID is defined as an integer, do not change it to a varchar. However, increasing the number of characters in a varchar column is an acceptable change.

The schemas also store path and file names to job scripts, providing for easy updates to several jobs if they share a script.

For enterprise applications, you need to use a supported commercial database. For information about setting up a commercial back-end database for use by Siebel Analytics Scheduler, see *Siebel Analytics Installation and Configuration Guide*.

NOTE: There is a one-to-one relationship between the back-end database and Siebel Analytics Scheduler. Do not configure multiple Siebel Analytics Scheduler applications to use a single back-end database.

Scheduler Log File

Information about Scheduler startup and shutdown activities is logged to the NQScheduler.log file, located in the Log directory in the Siebel Analytics software installation directory. You can use a text editor to view this file. The entries in the log file are written in UTF-8 format. Set your viewer to UTF-8 to display readable text.

Installation, Upgrade, and Configuration Overview for Analytics Scheduler

The following topics provide an overview of how to install, upgrade, and configure Siebel Analytics Scheduler.

- "High-Level Installation and Configuration Procedure for Analytics Scheduler" on page 13
- "Upgrading Analytics Scheduler from Version 7.5" on page 13
- "Configuring Siebel Analytics Scheduler" on page 14

For detailed installation information, see *Siebel Analytics Installation and Configuration Guide*.

High-Level Installation and Configuration Procedure for Analytics Scheduler

The following process provides the high-level steps to install and configure the Siebel Analytics Scheduler.

1. Install Siebel Analytics Scheduler.
2. If you are not using Siebel eBusiness Applications OLTP, set up database tables.
3. Start Siebel Analytics Server.
4. Populate configuration options:
   - **Windows**. Open Siebel Analytics Administration Tool, and then open the Job Manager and populate configuration options.
   - **UNIX**. Execute the file `schconfig.exe` on the UNIX machine hosting Siebel Analytics Scheduler. This file is located in the INSTALLDIR/Bin directory.
5. Start Siebel Analytics Scheduler.
6. Verify Siebel Analytics Web Server points to Siebel Analytics Scheduler.

NOTE: You may need to configure Siebel Analytics Server registry settings. The following registry entry identifies the name of the machine running the Scheduler:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Siebel Systems, Inc.\Siebel
Analytics\Web\n.n\Alerts

ScheduleServer
```

where `n.n` is the version number of Siebel Analytics. For more information, see *Siebel Analytics Web Administration Guide*.

Upgrading Analytics Scheduler from Version 7.5

To upgrade from Siebel Analytics, Version 7.5, first install Siebel Analytics Version 7.7.

- If you upgrade to Siebel Analytics Version 7.7 and currently use Siebel eBusiness Applications Version 7.5.x at the same time, then you do not need to upgrade the database schema separately. The schema upgrade should have been done in the installation procedure outlined in the *Siebel Analytics Installation and Configuration Guide*. That installation procedure includes a step to import a Siebel Import File (SIF) that modifies the Scheduler database schema.
- If you install the stand-alone version of Siebel Analytics 7.7, then you must upgrade the database schema for Analytics by running the following file:
  `SAJOBS.7.5To7.7.xxx.sql`
  Where `xxx` refers to your RDBMS type: DB2, MSSQL, or Oracle.

  If you get an error indicating that the TZ_NAME column is not found, shut down the Scheduler and rerun the `SAJOBS.7.5To7.7.xxx.sql` file.

Configuring Siebel Analytics Scheduler

In Windows you set Scheduler configuration options in the Job Manager Configuration dialog box, a feature of Siebel Analytics Administration Tool. This dialog box has three tabs: Scheduler, Siebel Mail, and Siebel iBots.

In UNIX you set Scheduler configuration options in `schconfig.exe`, a console-based application. The configuration options in `schconfig.exe` are identical to those available in the Job Manager Configuration dialog box.

This section provides detailed information about the Scheduler configuration options.

To access the Job Manager Configuration dialog box in Windows

1. Open the Siebel Analytics Administration Tool.
2. Choose Manage > Jobs from the toolbar.

   The Job Manager window appears.
3. Choose File > Configuration Options.

   The Machine Name dialog box appears.

4 Enter the machine name where the Scheduler is located, and then click OK.

The Job Manager Configuration dialog box appears.

Job Manager Configuration—Scheduler Tab

The fields in the Scheduler tab describe Siebel Analytics Scheduler access to its back-end database and its general behavior. To restore the default values distributed with Scheduler, click Defaults. To revert to the most recently used settings, click Revert.

Table 3 provides a description of each field in the Scheduler tab.

Table 3. Scheduler Tab Fields

Field	Description
Bulk Fetch Buffer Size (bytes)	This parameter is for use in the database gateways. It specifies the maximum size in bytes of a bulk fetch page for retrieving data from a data source. The default value is 33,792 bytes.
Call Interface	The type of call the connection to the database makes. The call you choose from the pull-down list determines the application programmer interface (API) used to access the data source. The types of call interfaces in the list varies depending on the database type selected.
Data Source Name	Identifies the data source name (DSN) of the data source to use for the connection. For example, if the DSN of the Scheduler back-end database is ORCL8_Scheduler, you would enter ORCL8_Scheduler in this field. The DSN must contain valid logon information for a data source. If the information is invalid, the database logon fails.
Database Type	The database type of the Scheduler back-end database. Choose a type from the drop-down list.
Default Script Path	This field refers to a path where user-created job scripts (not iBots) should be stored. In the Job Manager, if a filename is entered in the Script field when adding or modifying a job, Siebel Analytics Scheduler examines the contents of this directory for the specified file. However, if a full path is given in the Script field, this directory is not examined. By default, this field is set to `$(SiebelAnalytics)\Scripts\Common` where `$(SiebelAnalytics)` is the location in which the Siebel Analytics software is installed.

Table 3. Scheduler Tab Fields

Field	Description
Maximum Connections	The maximum number of database connections the Scheduler can open concurrently. When this limit is reached, the connection request waits until a connection becomes available. The default value is 5. Specify a value of 1 or greater.
Maximum Execution Threads	This field specifies the maximum number of threads that are used in Siebel Analytics Scheduler's thread pool.
Minimum Execution Threads	Siebel Analytics Scheduler is a multithreaded application. It uses a thread pool. This field specifies the minimum number of threads in this pool.
nQS (Siebel) Administrator Name	Because Siebel Analytics Scheduler runs jobs on Siebel Analytics Server and Siebel Analytics Web on behalf of a user without storing that user's password, the server allows the Scheduler and the Web to impersonate users. To allow these applications to impersonate users, specify a user with administrator's privileges in this field. The user specified must be a Repository Administrator with the ability to impersonate other users, and have basic access to the Web catalog (to access the shared folder).
nQS (Siebel) Administrator Password/ Confirm Password	These fields contain the password for the Siebel administrator with impersonation privileges. The password is encrypted in the registry. You can change it by using the Job Manager (Windows) or `schconfig.exe` (UNIX).
Password/ Confirm Password	These fields contain the password to use to log on to the data source. For security, the password is encrypted in the registry.
Pause When Service Starts	This is a flag that specifies that no jobs should execute when Siebel Analytics Scheduler starts. While it is paused, users can add, modify, and remove jobs. However, no jobs execute. The Siebel administrator can use this field for maintenance purposes, such as shutting down the Scheduler, then setting this value, and then restarting the Scheduler to conduct maintenance. Choose Service Management > Continue Scheduling in the Job Manager to continue with regular execution.

Table 3. Scheduler Tab Fields

Field	Description
Purge DB every X minutes	Depending on the back-end database, deleting rows from a table may be expensive, while updates are almost always streamlined. Siebel Analytics Scheduler does not actually issue SQL DELETE statements when jobs or instances are removed. Instead, it updates the tables and flags the affected rows as deleted. After every X minutes (where X is defined as the value of this field), the actual SQL DELETE statements are issued. The default value is every 60 minutes.
Purge Old instances after X days	Specifies the number of days after which old job instances are deleted from the back-end database automatically. The default value is to delete after seven days. To prevent old job instances from being deleted automatically, set the value to 0 (zero).
Scheduler Script Path	The Scheduler's back-end database does not actually store the job scripts. This field refers to the path where Scheduler-created job scripts are stored. In the Job Manager, when adding or modifying a job, if a script is manually entered in the Script field, the Scheduler creates a file in the path defined by this field. The file has an SCS extension. In general, you should not add or remove scripts from this directory. By default, this field is set to `$(SiebelAnalytics)\Scripts\Scheduler` where `$(SiebelAnalytics)` is the location in which the Siebel Analytics software is installed.
Temporary File Path	This is a path that specifies where temporary files are stored during the execution of Siebel Analytics Scheduler.
Timeout (Minutes)	The timeout specifies the amount of time, in minutes, that a connection to the data source remains open after an operation completes. During this time, new operations use this connection rather than open a new one, up to the number specified for Maximum Connections. The time is reset after each completed connection request. The default value is 60. Specify a value of 1 or greater.
User name	The user name the Scheduler uses to log on to the data source. The user name must have read and write permissions to the back-end database. To properly set up Scheduler, the user name must be the same as the table owner for databases like DB2 that require fully qualified table names (for example, SIEBEL.S_NQ_JOB).

Job Manager Configuration—Siebel Mail Tab

The fields in the Siebel Mail tab are used to configure the COM object used by Siebel Delivers to deliver SMTP mail. To restore the default values distributed with Siebel Analytics Scheduler, click Defaults. To revert to the most recently used settings, click Revert.

Table 4 provides a description of each field in the Siebel Mail tab.

Table 4. Siebel Mail Tab Fields

Field	Description
Authenticate against SMTP Server	This specifies that the SMTP Server requires authentication to send email to an address outside of its domain. When checked, fill in the Username and Password fields for an email user on the SMTP server. When not checked, the Username and Password fields are not used.
From Display Name	This field is used in the SMTP From field as a meaningful substitution for the sender's address. The default is `Siebel Delivers <iBots@defaultmailserver.com>`.
Maximum Recipients	Some mail servers limit the number of SMTP recipients for a single email. Usually this limit is close to 1000 recipients. Use the Maximum Recipients field to prevent sending to more recipients than your mail server's limit. If the mailing list has more subscribers than the Maximum Recipient value, Siebel Delivers splits the list into smaller lists and sends an email for each list.
Number of Retries Upon Failure	If Siebel Delivers fails to deliver an email, it tries to resend it. The number specified in this field sets the number of retry attempts. The default is 1.
Password/ Confirm Password	When the option Authenticate against SMTP Server is checked, these fields specify the user's password for the specified user name.
Sender Address	This is the email address on the SMTP Server used as the sender's reply-to address for all mail sent from Siebel Analytics Scheduler. The initial value is `defaultuser@defaultmailserver.com`, which must be changed to reflect a valid email address.
SMTP Port	This is the port number for SMTP on the given server. The default is port 25. Most mail servers use port 25 as the mail port.
SMTP Server	This is the name of the SMTP server that delivers the mail. The initial value is `defaultmailserver.com`, which must be changed to a valid server name in order to deliver mail.

Table 4. Siebel Mail Tab Fields

Field	Description
Use Bcc Instead of To	Use this option to put the subscribers list on the BCC line instead of the To line. This prevents subscribers from seeing the names of the others on the list, and it prevents subscribers from accidently (or maliciously) replying to the entire list.
Username	When the option Authenticate against SMTP Server is checked, this field specifies the user name for an email user on the specified SMTP server. If left blank, the email server does not authenticate any email from Siebel Delivers.

Job Manager Configuration—Siebel iBots Tab

Siebel iBots are functionally a combination of data stored in Siebel Analytics Web and on Siebel Analytics Scheduler. The fields in the iBots tab describe the behavior of all Siebel iBots that run on a specific Siebel Analytics Scheduler. To restore the default values distributed with the Scheduler, click Defaults. To revert to the most recently used settings, click Revert.

Table 5 provides a description of each field in the iBots tab.

Table 5. iBots Tab, Configuration Fields

Field	Description
Debug Enabled	Check this to have the Scheduler generate a log file for each iBot. This log file has useful logging messages when trying to diagnose a problem.
iBot Log Directory	Siebel iBots have the ability to create log files if exceptional error conditions occur. This setting specifies the directory where these files are saved. The directory must be accessible to the Scheduler application. In Windows, the default installation runs the service as a system account, which prevents Siebel Analytics Scheduler from writing to or reading from network directories.
Max Concurrent Subscribers	Specifies how many concurrent actions or unique requests can be executed in parallel. This number represents the number of dedicated threads from the thread pool an iBot uses. You can override this setting for individual iBots in the Modify Job dialog box.

Table 5. iBots Tab, Configuration Fields

Field	Description
Minimum/ Maximum Delivery Sleep Seconds	The minimum and maximum number of seconds during which the iBot randomly sleeps before it attempts to reconnect to the server (if its connection was refused) to deliver results.
Minimum/ Maximum Global Sleep Seconds	The minimum and maximum number of seconds during which the iBot randomly sleeps before it attempts to reconnect to the server (if its connection was refused) to get global information.
Minimum/ Maximum Request Sleep Seconds	The minimum and maximum number of seconds during which the iBot randomly sleeps before it attempts to reconnect to the server (if its connection was refused) to issue requests.
Number of Delivery Tries	After a unique request has executed, the iBot tries to deliver the results to specified devices. This field specifies the number of times the Scheduler attempts to connect to the server to deliver the results.
Number of Global Tries	When a Web or mail server has too many people logged on, the server may reject new connections, including connections from Siebel Analytics Scheduler. To cope with this, An iBot retries the connection. This field sets the maximum number of tries to get global information (what to deliver and to whom) before the iBot gives up.
Number of Request Tries	After an iBot has received the global information, it issues a series of unique requests to the server for each user. This field specifies the number of times the Scheduler attempts to connect to the server to issue these requests.
SAW Machine Name	The machine name and port where SAWServer is running. The server name must be specified and can optionally be followed by a colon and port number. If you do not specify a port number, Scheduler uses the default SAW server port number, which is 9710. For example, if the SAWServer runs on machine1 and port 8900, the value for this field is machine1:8900.

Job Manager Configuration—Workflow Tab

Siebel Delivers can trigger Siebel eBusiness Applications V7.5 workflows. For information on how to set this up, see "Integrating Workflows with Siebel Delivers" on page 34. Part of the process of requires using the controls on the Workflow tab.

Table 6 provides a description of each field in the Workflow tab.

Table 6. Workflow Tab, Configuration Fields

Field	Description
Server	The name of your Siebel enterpriser server. Siebel Analytics Scheduler uses the http connector of the server to trigger the workflow. Example: http://localhost
System	The name of the Siebel EAI system in your environment. Example: eai_enu
Service	The name of the http service as defined in the eai.cfg file. Example: ANALYTICS
User	The user name the Scheduler uses to log on to the Siebel Enterprise Server.
Password	Password for the user account.

Scheduler as a Background Process

Starting the Scheduler as a background process requires that you export the display to an XServer that is always running. Then run the initialization script followed by the Scheduler executable. The following example is for bash shell.

```
export DISPLAY="myaccount:0.0"
sa.sh
nqscheduler.exe &
```

You may need to modify this example for other shells or to use DISPLAY options.

About the Analytics Job Manager

To schedule jobs to run at any time, use the Job Manager feature of Siebel Analytics Administration Tool. You can set options for a start time, a start date, an interval between executions, and an optional end time and date. For information about scheduling Siebel iBots, see *Siebel Analytics Web Administration Guide*.

The Job Manager is the interface to Siebel Analytics Scheduler. When opened, the Siebel administrator can connect to and configure Siebel Analytics Scheduler, start and stop the Scheduler, add and manage jobs, and manage job instances.

Job Manager Toolbar Options

This topic describes the options available in the various menus on the Job Manager toolbar. Table 7 provides a description of the File menu options.

Table 7. File Menu Options

Field	Description
Open Scheduler Connection	Opens the Machine Name dialog box, where you specify the name of the machine running Siebel Analytics Scheduler. If the Scheduler resides on the same machine as the Administration Tool, you can enter `local`.
Close Scheduler Connection	Closes the Job Manager connection to Siebel Analytics Scheduler.
Configuration Options	Opens the Machine Name dialog box, where you specify the name of the machine running Siebel Analytics Scheduler, and then the Job Manager Configuration dialog box appears, where you can set configuration options.
Exit Job Manager	Shuts down the Job Manager and returns you to the Administration Tool. If you exit the Job Manager while a connection to the Scheduler is still open, the connection closes.

Table 8 provides a description of the Service Management menu options.

Table 8. Service Management Menu

Field	Description
Pause Scheduling	Stops all jobs from executing until scheduling is continued. This is sometimes required for maintenance purposes. If the Scheduler is stopped while scheduling is paused, scheduling continues when it is restarted, unless the option Pause When Service Starts is set in the Scheduler tab of the Job Manager Configuration dialog box.
Continue Scheduling	Resumes the Scheduler's regular execution.
Stop Service	Stops the Scheduler service.

Table 9 provides a description of the Jobs menu options.

Table 9. Jobs Menu

Field	Description
Add New Job	Open the Add Job window, where you specify the properties for a new job.
Remove Job(s)	Removes the selected job or jobs from the Scheduler. When a job is removed, all instances for that job are removed as well.
Modify Job	Open the Modify Job window, where you can modify the properties for an existing job.
View Job	Opens the View Job window, where you can view the properties for a job.
Refresh Job List	Refreshes the job information displayed in the Job List in the right pane.

Table 10 provides a description of the Instances menu options.

Table 10. Instances Menu

Field	Description
Cancel Instance(s)	Cancels the selected running instance.
Purge Instance(s)	Opens the Purge Instances dialog box, where you can specify the purge method to use.
View Instance	Displays information about the selected instance.
Refresh Instance	Refreshes the instance information displayed in the Instance List in the right pane.

Setting Up Analytics Scheduler Jobs

A job has properties, one or more actions to perform, and an execution schedule that determines when it runs. A Siebel administrator can add a job through the Job Manager, but jobs are most commonly added through Siebel Delivers.

NOTE: You cannot add iBot jobs using the Job Manager. The Siebel Analytics Server passes the iBots to the Scheduler.

You can, however, modify iBots using the Modify Job dialog box. For more information, see "Modifying iBots" on page 31.

Use the Add Job and Modify Job dialog boxes in the Job Manager to add and modify jobs. These dialog boxes contain three types of information: general job properties, a Script area where the job actions are specified, and a Trigger area where the job trigger is specified.

To add a job

1. In the Job Manager, choose Jobs > Add New Job.
2. Enter the appropriate information in the dialog box.

   Refer to the following for field descriptions:

   - "Setting Up and Modifying General Scheduler Job Properties" on page 25
   - "Defining Scheduler Job Actions" on page 27
   - "Setting Scheduler Job Triggers" on page 28

To modify a job

- In the Job Manager, select the job you want to modify, and then choose Jobs > Modify Job.

  The Modify Job dialog box appears, where you can change job properties.

Setting Up and Modifying General Scheduler Job Properties

Use the following fields in the Add Job or Modify Job dialog box to configure or modify the general properties for a job.

Table 11 provides a description of the general job properties available in the Add Job and Modify Job dialog boxes.

Table 11. General Scheduler Job Properties

Field	Description
JobID	This display-only field becomes visible when choosing Jobs > View Job or Jobs > Modify Job. After a job has been added, this field contains an integer that uniquely identifies the job. The JobID is generated internally and cannot be changed. This field is also included in the Job List display in the right pane of the Job Manager window.
Name	Enter a short, descriptive name for the job. This field is also included in the Job List display in the right pane of the Job Manager window.
Description	Enter a brief description of the job that describes its actions to end users. This field is also included in the Job List display in the right pane of the Job Manager window.
UserID	This field is required for all jobs. For jobs that communicate with Siebel Analytics Server or Siebel Analytics Web, this needs to be a valid Siebel Analytics user ID. When this job runs, the Scheduler executes it on behalf of the user ID specified in this field. This field is also included in the Job List display in the right pane of the Job Manager window.
Maximum Run Time MS	This specifies the maximum number of milliseconds this job should run before it is canceled forcibly. If a job exceeds its run time, it fails with a “time out” reason code. To prevent the job from timing out, set this field to 0 (zero).
Last Run Time	This display-only field shows the last time this job began execution. This field is also included in the Job List display in the right pane of the Job Manager window.
Next Run Time	This display-only field is for recurrent jobs. It shows the next time this job executes. The trigger is used to determine this value.
Running Instance Count	This display-only field shows the number of currently running instances of this job.

Table 11. General Scheduler Job Properties

Field	Description
Delete Job When Done	When this option is checked, Scheduler deletes the job after its last scheduled execution as defined by its trigger. When there is no next run time, the job is done. When a job is deleted, all instances are deleted as well. For most jobs, this option should not be checked, because deleting a job can be done manually through the Job Manager.
Disabled	When this option is checked, the job script does not execute when its trigger expires. However, the next run time is still updated according to the trigger settings. This is useful when testing or debugging a new job because a Siebel administrator can quickly disable a job without losing all information.
Execute When Missed	When this option is checked, and Siebel Analytics Scheduler is stopped (either all scheduling is paused or the Scheduler application is stopped), the job runs when Siebel Analytics Scheduler is brought back up if its next run time was missed. When not checked, the job executes at its next run time defined by its trigger.
Delete Script When Job is Removed	If this option is checked, when a job is removed its associated job script is also removed. If many jobs reference the same job script, this option should not be set.
Disable NQS Functions	This option disables the NQS Connect function in the job script. If a job script is not trusted by the Siebel administrator, this option can be checked to ensure that the user does not try to connect to Siebel Analytics Server through impersonation methods. In most cases, as security is checked by Siebel Analytics Web and the Administration Tool, or when jobs are added by the Siebel administrator, this option does not need to be set.

Defining Scheduler Job Actions

Use the following fields in the Script area of the Add Job or Modify Job dialog box to define the actions a job performs when it executes. Table 12 provides a description of the job action properties available in the Add Job and Modify Job dialog boxes.

Table 12. Job Action Fields

Field	Description
Script Type	Currently, two types of scripts are supported by Siebel Analytics Scheduler: VBScript and JScript. This field is set according to the type of script referred to by the Script field.
Script Contains File Name	In the Job Manager, you can enter either a file name or the actual contents of a script in the Script field. If the Script field contains a file name, this option is checked.
Script	This is either a reference to a job script file or the contents of a job script itself. If it is a reference, enter a file name in this field, such as TestConnect.js. If no path is given, Siebel Analytics Scheduler examines the directory referred to in the Default Script Path configuration value in the Scheduler tab of the Job Manager Configuration dialog. If a path is given, the path must be accessible by the Scheduler application. In Windows, the default Siebel Analytics Server installation registers the service as a system account, which makes network paths inaccessible.

Parameters	The values entered in this field are passed to the job script through the Parameters array. Enter one parameter per line. For example:
	`c:\siebel\data\scheduler`
	`cli_snowflake`
	`SELECT Lastname FROM Employee`
	By default, each parameter must be less than 255 characters in length. You can change this restriction by modifying the varchar length in the back-end database. In the NQSJOBPARAMETERS table, the column is JobParameter. For more information about this table, see *Siebel Analytics Installation and Configuration Guide*.

Setting Scheduler Job Triggers

A job trigger determines when and how often it executes. Use the following fields in the Trigger area of the Add Job or Modify Job dialog box to define the actions a job performs when it executes.

There are two types of triggers: single-run triggers and recurrent triggers.

Single-Run Triggers

Use the Trigger Type drop-down list to choose the trigger type. There are two single-run triggers, Run Now and Run Once.

Run Now

This trigger specifies that the job runs immediately. It executes only one time.

Run Once

Jobs of this trigger type execute at the date and time specified in the Begin Date and Start Time fields, which become active when you choose Run Once. An error occurs if the given time is in the past. If the option Set Start Time To Now is checked, this trigger is equivalent to the Run Now trigger.

Recurrent Triggers

All recurrent triggers specify that the job execute over a period of time at given intervals.

The fields described in Table 13 pertain to all recurrent triggers.

Table 13. Recurrent Trigger Fields

Field	Description
Begin Date	This specifies the date when the first recurrent interval runs. The recurrent interval is defined as the time between Start Time and End Time. This field is hidden if the option Set Start Time to Now is checked.
End Date	This field becomes active when the option Has End Date is set. This specifies the date when the last recurrent interval is run. If no end date is set, the job runs “forever.”

Table 13. Recurrent Trigger Fields

Field	Description
Start Time	This specifies the lower bounds of the recurrent interval. The job's first execution for a day occurs at the time specified in this value.
End Time	This specifies the upper bounds of the recurrent interval. The job's last execution for a given day occurs at or before the time specified in this value. If this value is less than the Start Time value, the interval spans midnight of the given day. For example, a trigger with a start time of 11:00 PM and an End Time of 2:00 AM starts its execution on the date specified in Begin Date at 11:00 PM and continues until 2:00 AM on the following day.
Has End Date	If this option is not checked, the job runs in perpetuity. If this option is checked, specify an End Date.
Set Start Time To Now	If this option is checked, the Begin Date and Start Time fields are ignored and their values are populated with Siebel Analytics Scheduler's current date and time.
Interval in Minutes	During the recurrent interval, this field specifies the number of minutes between subsequent executions of a job. A job starts execution promptly at its Start Time, and executes again every *n* minutes, where *n* is the value of this field.
Maximum Concurrent Instances	If a job executes every *n* minutes (from the Interval in Minutes field), a long-running job may have overlapping executions. Use this field to set the number of concurrent running instances. For an unlimited number of concurrent instances, set this value to zero.

Recurrent Trigger Types

The fields described in Table 14 are the recurrent triggers available from the Trigger Type drop-down list. Depending on the trigger type you choose, additional options become active. Refer to the examples to see how these additional options can be used.

Table 14. Recurrent Trigger Types

Type	Description
Daily	Runs a job every day or every few days. The Days Interval field specifies the number of days between each subsequent recurrent interval. **Examples:** To run a job every hour between 8:00 AM and 5:00 PM starting on January 1, 2001 and ending on January 15, 2001, set the Begin Date to 1/1/01, the Start Time to 8:00 AM, and the End Time to 5:00 PM. Set the Has End Date flag, the End Date to 1/15/01, the Interval in Minutes to 60, and the Days Interval to 1. To run a job every five minutes forever, set the Begin Date to the desired date, the Start Time to 12:00 PM, the end time to 11:59 AM, the Interval In Minutes to 5, and the Days Interval to 1.
Weekly	Runs a job on specified days of the week. The Weeks Interval specifies the number of weeks between each execution. The Days of the Week field specifies on which days the execution occurs. **Example:** To run a job at noon every other week on Mondays, Wednesdays, and Fridays, set the Begin Date to the desired date, the Start Time and End Time to 12:00 PM, the Interval in Minutes to 1, the Weeks Interval to 2, and the Days of the Week to Monday, Wednesday, and Friday.

Table 14. Recurrent Trigger Types

Type	Description
Monthly by Date	Runs a job on specific days of the month. The Months field specifies in which months this job executes. The Days field specifies which days of those months. If the given day does not exist for a given month, that day is ignored. **Examples:** To run a job at 5:00 PM on the 1st and 15th of January, February, and March, set the Begin Date to January 1, the Start Time and End Time to 5:00 PM, the Interval in Minutes to 1, the Months to January, February, and March, and the Days to 1 and 15. To run a job at 2:00 AM on every leap day (February 29th), set the Begin Date to January 1, the Start Time and End Time to 2:00 AM, the Interval in Minutes to 1, the Months to February, and the Days to 29.
Monthly by DOW	Runs a job on specific occurrences of specified days of the week during given months. The Months field specifies which months this job executes. The Days of the Week field specifies which days of the week the job executes during those months. The Occurrence field specifies which of those days to execute. The occurrence can be any or all of First, Second, Third, Fourth, and Last. The Last value specifies that either the fourth or fifth occurrence of a given day is used, depending on whether there are four or five occurrences during that month. **Examples:** To run a job on the first and third Fridays of December every hour between the hours of 4:00 AM and 8:00 PM, set the Begin Date to the desired date, the Start Time to 4:00 AM, the End Time to 8:00 PM, the Interval in Minutes to 60, the Months to December, the Days of the Week to Friday, and the Occurrence to the First and the Third. To run a job at 3:00 AM every time that Daylight Saving Time switches over to Standard Time, set the Begin Date to the desired date, the Start Time and End Time to 3:00 AM, the Months to October, the Days of the Week to Sunday, and the Occurrence to Last.

Modifying iBots

You can modify individual iBots using the Modify Job dialog box in the Job Manager, accessible from Siebel Analytics Administration Tool.

To modify an iBot

- In the Job Manager, select the iBot you want to modify, and then choose Jobs > Modify Job.

  The Modify Job dialog box displays the iBot you selected. Modify the iBot properties, as described in Table 15.

  A default value in these fields indicates that the value specified in the Siebel iBots tab of the Job Manager Configuration dialog box is active.

Table 15. iBot Properties

iBot Property	Description
Web Server	Specifies the Web server this iBot contacts when it runs. Do not change this setting, because the iBot may not exist on a different Web Server. This feature was added for debugging purposes only.
Debug Log	Determines whether debugging information is written to a log.
Max Concurrent Subscribers	Specifies how many concurrent actions or unique requests are executed in parallel. **NOTE:** This number represents the number of dedicated threads from the thread pool that this iBot uses. If you have many subscribers, and the iBot is personalized for each recipient, it may be beneficial to increase the number of threads for this iBot, because it can handle more subscribers concurrently. However, you should exercise caution, because if a subscriber receives multiple deliveries, threads can become blocked. For example, if you have two subscribers, each with a delivery to an email device and to the Intelligence Dashboard, then two threads can execute concurrently while the other two wait for the results of the first two. If you have allocated five threads to this iBot, all deliveries try to execute, but two occupy threads that could be used by other iBots. The other iBots are not able to execute if one long-running iBot uses all of the threads. If many iBots are scheduled to execute at the same time, the cumulative number of threads you dedicate (that is, the sum of all iBots' threads) should not be greater than the current thread pool. Otherwise, performance could be impeded.

Working with Scheduler Job Instances

An instance is a record that stores information regarding a specific execution of a job.

To work with job instances, click the Instances tab in the lower-left corner of the Job Manager window. When instances are present, you can use the tree in the left pane to locate instances and view information about them.

Managing Instances

- To cancel an instance, select it and choose Instances > Cancel Instance(s). The cancel event is issued to Siebel Analytics Scheduler and the instance is marked as canceled when its registered cancel methods are called. For more information about registered cancel methods, see the command "RegisterCancelCommand Method" on page 46.
- To view information for a particular instance, select it and choose Instance > View Instance. This opens the Instance window.
- Purging a job instance involves removing it from the back-end database. You can purge a job instance in either of these ways:
  - To delete one or more instances, select them from the Instance List and press Delete.
  - Click the Purge Instance(s) icon on the toolbar or choose Instances > Purge Instances to open the Purge Instances window. You can purge instances by JobID, by UserID, or by End Time. If you choose the End Time method, all jobs with an End Time less than or equal to the given time are purged. Choose the purge method to use, and click OK when you are done to return to the Job Manager window.
- To view information for a particular instance, either double-click it, or select it and choose Instances > View Instance. This opens the Instance window.
- To refresh the information in the Instance List, choose Instances > Refresh Instance List. In some environments, if numerous instances have run and instances have not been purged in some time, this may take a few seconds.

Integrating Workflows with Siebel Delivers

Siebel Delivers supports native integration with Siebel Workflow Version 7.5 so that iBots can trigger workflow tasks. This requires that the workflow tasks be already setup in the Siebel eBusiness applications. For more information, read *Siebel Business Process Designer Administration Guide*, Version 7.5.

There are two parts to setting up iBots to trigger workflow.

1. The administrator configures the Scheduler as described in "Process of Configuring Siebel Delivers to Trigger Workflows." This configuration provides the Scheduler with the location of the Siebel Enterprise Server.
2. Create workflow triggers in the iBots. For more information, see the *Siebel Analytics User Guide*.

Process of Configuring Siebel Delivers to Trigger Workflows

To configure Siebel Delivers to launch workflow applications, perform the following tasks.

- To configure the Siebel Enterprise Server for Workflow integration with Siebel Delivers
- To configure Siebel Analytics Scheduler to trigger workflows on page 35

To configure the Siebel Enterprise Server for Workflow integration with Siebel Delivers

This task is a step in Process of Configuring Siebel Delivers to Trigger Workflows on page 34.

1. On the Siebel Enterprise Server, create a named subsystem specific to Analytics using the SrvrMgr command line interface using the following command. Replace `<subsystem_name>` with a name you create, such as AnalyticsWFDispatch.

```
create named subsystem <subsystem_name> for subsystem
EAITransportDataHandlingSubsys with ConverterService="XML
Converter",DispatchService="Workflow Process
Manager",DispatchMethod="RunProcess"
```

2 Open the eai.cfg configuration file inside the SiebSrvr directory and add the following line under the http services section.

```
[Http Services]
SiebelQuery = SiebelQueryDispatch
SiebelUpsert = SiebelUpsertDispatch
SiebelExecute = SiebelExecuteDispatch
ANALYTICS = <subsystem_name>
```

3 Restart the Siebel Server.

To configure Siebel Analytics Scheduler to trigger workflows

This task is a step in Process of Configuring Siebel Delivers to Trigger Workflows on page 34.

1 In the Siebel Analytics Administration Tool, open the Job Manager.

2 Choose File > Configuration Options.

The Job Manager Configuration window opens.

3 Click the Workflow tab.

4 Fill out the fields using the information in "Job Manager Configuration—Workflow Tab" on page 20.

External Triggering of Scheduler Jobs

Third-party applications and scripts can launch a Scheduler job from the command line. They can also change the Job Parameters for a single instance. This simulates third-party triggered iBots. The interface for the command line is:

```
saschinvoke.exe –u <Admin Name>/<Admin Password> (-j <job id> |
-i <iBot path>) [-m <machine name>[:<port>]] ([-r <replace
parameter filename>] | [-a <append parameter filename>])
```

The required <Admin Name> and <Admin Password> are the same as configured for the Scheduler where you invoke the job. You can invoke the job either by the job id or by the iBot path. Optionally, you can specify a machine and port for the Scheduler. If this is omitted, the invoker uses localhost and 9705 respectively.

The invoker also takes an optional job parameter file. Depending on the option, the parameters in the file either replace the ones configured in the Scheduler (`[-r <replace parameter filename>]`, or they append to the existing parameters `[-a <append parameter filename>]`).

If you use a parameter file, follow these rules.

- Only one parameter per line.
- White space is not ignored because it may be custom script dependent.

In replace mode, the file can specify to leave some parameters as they are in the Scheduler. To do this for a specific line, enter $SCH_DEFAULT$ on the line. This acts as a variable and replaces the $SCH_DEFAULT$ text with the original text from the parameter. For example, if the original parameter is "hello", the line "$SCH_DEFAULT$ world, $SCH_DEFAULT again" would be changed to "hello world, hello again". It is the caller of saschinvoke's responsibility to correctly pass in parameters. The executable does not test the parameters for correctness.

Working with Scheduler Job Scripts

Siebel Analytics Scheduler supports two types of scripting languages that can be used for job scripts: VBScript and JScript. Any VBScript or JScript can be used as a job script. The Scheduler has language extensions to provide these scripts with additional functionality when defined within the context of Siebel Analytics.

Siebel Analytics Scheduler includes a Script object that encapsulates a running script. The Script object represents a script and exposes the properties and methods of a script. You can access its methods and properties directly because its name is implied. For example, to access the JobID property, you can specify JobID not Script.JobID.

For detailed information on working with job scripts, read the following topics.

- "Read-Only Script Object Properties" on page 37
- "Read/Write Script Object Properties" on page 38
- "Script-Defined Constants" on page 39
- "Script Object Methods and Events" on page 43

Read-Only Script Object Properties

Siebel Analytics Scheduler supports the following read-only script object properties.

JobID Object Property

Returns the job identification number associated with this instance.

Return Value Returns a long value.

InstanceID Object Property

Returns the instance identification number associated with this instance.

Parameter Count Object Property

Returns the number of job parameters associated with the job script.

Return Value Returns a long value.

Parameter (index) Object Property

Returns a specific parameter associated with the script. Parameter (index) returns an error if the given index is less than zero or greater than Parameter Count minus 1.

Syntax

```
Parameter(index)
```

Arguments

Argument	Description
index	The zero-based index of the parameter.

Return Value Returns a string value.

Script Object Property

Returns the Script object that represents the current script. This object implements the COM IDispatch interface and can be passed as arguments to methods of other COM objects that exist on the system. This is particularly useful when handling cancel events to a running instance. See "RegisterCancelCommand Method" on page 46.

Return Value Returns a script object.

Return Value Returns a double value.

UserID Object Property

Returns the user identification number associated with the instance.

Return Value Returns a string value.

Read/Write Script Object Properties

Siebel Analytics Scheduler supports the following read-only script object properties.

- "Error Message Object Property" on page 38
- "ExitCode Object Property" on page 39

Error Message Object Property

Sets or returns the Error Message property of the running instance.

In UNIX environments, the JScript throw() construct is not supported. Instead, the ErrorMessage property can be used to convey meaningful error information. Setting this value changes the Error Message field of a Job Instance. However, unlike throw(), it does not stop execution of the current Job Script.

In Windows environments, if the JScript throw() method is called and this property has been set, the value is appended to the message description in the JScript or VBScript Error object.

COM objects that implement the IDispatch interface can be accessed from within Job Scripts. If any method fails and properly provides error information through the SetErrorInfo() method, that information is contained in the Error Message field of the Job Instance. If the ErrorMessage property is set before the COM object error is generated, then that string value is appended to the COM object error information.

Be aware that if this property is set to a nonempty string value, the job instance has a Failed status and its ErrorMessage property is set. Resetting this value to the empty string ("") clears the error messages.

Settings and Return Value: sets or returns a string value.

ExitCode Object Property

Sets or returns the Exit Code property associated with the instance.

Settings and Return Value: sets or returns a long value. The default is 0 (zero).

Script-Defined Constants

Siebel Analytics Scheduler supports the following script-defined constants. These constants are used by the methods to schedule new jobs.

DayEnum Constants

The DayEnum values are used with the scheduling functions to identify days in a month, from Day 1 to Day 31. Table 16 describes DayEnum values.

Table 16. DayEnum Values

Constant	Description
nqDay1	Day 1
nqDay2	Day 2
nqDay3	Day 3
…	…
nqDay31	Day 31

DayOfWeekEnum Constants

The DayOfWeekEnum values are used with the scheduling functions to identify days in a week. Table 17 describes DayOfWeekEnum values.

Table 17. DayOfWeekEnum Values

Constant	Description
nqSunday	Sunday
nqMonday	Monday
nqTuesday	Tuesday
nqWednesday	Wednesday
nqThursday	Thursday
nqFriday	Friday
nqSaturday	Saturday

JobFlagsEnum Constants

The JobFlagsEnum values are used with the scheduling methods of the Script object to control how a job behaves. Table 18 describes JobFlagsEnum values.

Table 18. JobFlagsEnum Values

Constant	Description
nqJobNoFlags	Job has no special behavior.
nqJobDeleteWhenDone	Job is deleted when there are no more scheduled run times.
nqJobDisabled	Job is disabled. This is useful for preventing a job from running at the scheduled time or times.
nqJobHasEndDate	Job has a valid end date.
nqJobExecuteWhenMissed	If for some reason Siebel Analytics Scheduler is down when the job is supposed to start, this flag indicates that the job should run when the Scheduler starts up again.
nqJobDeleteScriptWhenDone	When a job is removed and this flag is set, the script associated with the job is deleted. This is useful only in conjunction with the nqJobScriptContainsPath flag.
nqJobScriptContainsPath	This flag indicates that the script associated with the job contains a path to a file containing the actual script code.
nqJobStartNow	When this flag is set, the begin date and start time are ignored. Instead, these fields get set to the current time of Siebel Analytics Scheduler.
nqJobIsUserScript	When scheduling another job from within a script, this flag specifies that the NQSConnect() function is disabled (the job is not a trusted job). This action effectively sets the nqJobIsUserScript flag for any job that the newly scheduled job may schedule.

MonthEnum Constants

The MonthEnum values are used with the scheduling functions to identify months. Table 19 describes MonthEnum values.

Table 19. MonthEnum Values

Constant	Description
nqJanuary	January
nqFebruary	February
nqMarch	March
nqApril	April
nqMay	May
nqJune	June
nqJuly	July
nqAugust	August
nqSeptember	September
nqOctober	October
nqNovember	November
nqDecember	December

OccurrenceEnum Constants

The OccurrenceEnum values are used with the scheduling functions to identify the occurrence of a given day. Table 20 describes OccurrenceEnum values.

Table 20. OccurrenceEnum Values

Constant	Description
nqFirst	First occurrence
nqSecond	Second occurrence
nqThird	Third occurrence

Table 20. OccurrenceEnum Values

Constant	Description
nqFourth	Fourth occurrence
nqLast	Last occurrence

Script Object Methods and Events

Siebel Analytics Scheduler has the following script object methods and Events available.

CreateArray Method

Creates an Array object.

Syntax 1

```
Set array = CreateArray ()
```

Syntax 2 `Set array = CreateArray ( size )`

Syntax 3 `Set array = CreateArray ( element0, element1, ..., elementn)`

Arguments

Argument	Description
size	A long value that specifies the initial size of the array.
element0 … elementn	The values to place in the array. This creates an array with the lower and upper bounds of 0 (zero) and n, respectively.

Return Value Returns an Array object.

Usage This method is provided only for JScript because local JScript Array objects cannot be passed directly to the Script methods. This method is called to create an array object and pass the array object to Script methods that accept an array as arguments.

The different syntax versions create arrays as follows:

- Syntax 1 creates an array of size 0 (zero).
- Syntax 2 creates an array with the specified size.
- Syntax 3 creates an array filled with the specified elements.

Example

```
var i;
var array1= CreateArray(2);
for (i = 0; i < array1.Size; i++)
{
   array1(i) = i;
}

   array1.Resize(4);
for (i = 2; i < array1.Size; i++)
{
   array1(i) = i;
}

var array2 = CreateArray(0, 1, 2,3);
for (i = 0; i < array2.Size; i++)
```

```
{
   if (array1(i) != array2(i))
   break;
}
```

DeregisterCancelCommand Method

Deregisters a previously registered cancel method.

Syntax `DeregisterCancelCommand`

Usage Call this method to deregister the most recently registered cancel method after a long operation has completed successfully. You do not need to call this method if the script was canceled.

GetConfigurationValue Method

Returns the value in Siebel Analytics Scheduler configuration relative to the Scheduler's root registry entry.

Syntax `value  = GetConfigurationValue(configKey [, subkeyPath])`

Arguments

Argument	Description
configKey	A string that specifies the registry key name to return.
subkeyPath	Optional. A string value that specifies the registry path below the Scheduler's root path.

Return Value Returns a string value.

Usage GetConfigurationValue() returns the string value for a registry setting relative to Siebel Analytics Scheduler. The `configKey` and `subkeyPath` strings must be identical to those in the registry.

GetTempFileName Method

Returns a temporary file name.

Syntax `tfname = GetTempFileName()`

Return Value Returns a string value.

Usage GetTempFileName() does not create a file. It only provides a temporary file name that can be used to create a file. Files created in job scripts are not deleted automatically when the script terminates.

LaunchProcess Method

Executes a command line in a new process.

Syntax `exitcode = LaunchProcess ( commandLine [, wait, terminateOnCancel] )`

Arguments

Argument	Description
commandLine	A string that specifies the command line to execute.
wait	Optional. A boolean value that specifies whether the method should wait for the process to terminate. The default is True.
terminateOnCancel	Optional. A boolean value that specifies whether the method should terminate the process when the script is canceled. The default is True.

Return Value Returns a long value.

Usage Call this method to execute a command line in a new process. If `wait` is set to True, this method returns the exit code returned by the process.

RegisterCancelCommand Method

Registers a method to be called when the script is canceled.

Syntax `RegisterCancelCommand source, methodName [, arguments]...`

Arguments

Argument	Description
source	An Object whose method is being registered.
methodName	A String that specifies the method name.
arguments	Optional arguments to be passed into the method.

Usage Occasionally, an object's method takes a long time to complete. If the job is canceled before the call returns, the script engine still must wait until the call returns. This could potentially take hours and tie up resources. This method solves the problem by registering a method that is asynchronously called by the script engine if the script gets canceled.

Cancel methods should be registered before calling the method that executes a long operation. When the method returns, the cancel method should be deregistered by calling DeregisterCancelCommand().

Good practice is to hide implementation details of a COM object from the caller, having the COM object itself handle all registration and deregistration of cancel commands. Pass an instance of the Script object to the COM object, then call the RegisterCancelCommand() and DeregisterCancelCommand() methods because the Script object implements the IDispatch interface.

ScheduleJobDaily Method

Schedules a new job with a Daily trigger.

Syntax

```
ScheduleJobDaily name, description, scriptType, script, startDate,
startTime, endTime, minutesInterval, daysInterval [, parameters,
flags, maxRunTimeMS, maxConcurrentInstances, endDate]
```

Arguments

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type associated with the job (either VBScript or JScript).

Argument	Description
script	A String that specifies the script code or path (if the nqJobScriptContainsPath flag is set) associated with the job.
startDate	A date value that specifies the date the job is activated.
startTime	A date value that specifies the time the job is activated.
endTime	A date value that specifies the time the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
daysInterval	An integer value that specifies the number of days between job invocations.
parameters	Optional. A string array of parameter values passed to the script. The default is an empty array.
flags	Optional. A long value that specifies the flags associated with the job. For valid settings, see "JobFlagsEnum Constants" on page 41. The default is nqJobNoFlags.
maxRunTimeMS	Optional. A long value that specifies the maximum time, in milliseconds, a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	Optional. A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	Optional. A date value that specifies the time the job is deactivated.

ScheduleJobMonthlyDate Method

Schedules a new job with a Monthly by Date trigger.

Syntax

```
ScheduleJobMonthlyDate name, description, scriptType, script,
startDate, startTime, endTime, minutesInterval, whichDays,
whichMonths [, parameters, flags, maxRunTimeMS,
maxConcurrentInstances, endDate]
```

Arguments

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) associated with the job.
startDate	A date value that specifies the date the job is activated.
startTime	A date value that specifies the time the job is activated.
endTime	A date value that specifies the time the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
whichDays	An long value that specifies the days of the month the job runs. For valid settings, see "DayEnum Constants" on page 40.
whichMonths	An integer value that specifies the months the job runs. For valid settings, see "MonthEnum Constants" on page 42.
parameters	Optional. A string array of parameter values passed to the script. The default is an empty array.
flags	Optional. A long value that specifies the flags associated with the job. For valid settings, see "JobFlagsEnum Constants" on page 41. The default is nqJobNoFlags.
maxRunTimeMS	Optional. A long value that specifies the maximum time, in milliseconds, a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	Optional. A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	Optional. A date value that specifies the time the job is deactivated.

ScheduleJobMonthlyDOW Method

Schedules a new job with a monthly by day of the week (DOW) trigger.

Syntax

```
ScheduleJobMonthlyDOW name, description, scriptType, script,
startDate, startTime, endTime, minutesInterval, whichOccurrences,
whichDays, whichMonths [, parameters, flags, maxRunTimeMS,
maxConcurrentInstances, endDate]
```

Arguments

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) associated with the job.
startDate	A date value that specifies the date the job activated.
startTime	A date value that specifies the time the job is activated.
endTime	A date value that specifies the time the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
whichOccurrences	An integer value that specifies the occurrences of days of week the job runs. For valid settings, see "DayEnum Constants" on page 40.
whichDays	An integer value that specifies the days of week the job runs. For valid settings, see "DayOfWeekEnum Constants" on page 40.
whichMonths	An integer value that specifies the months the job runs. For valid settings, see "MonthEnum Constants" on page 42.
parameters	Optional. A string array of parameter values passed to the script. The default is an empty array.
flags	Optional. A long value that specifies the flags associated with the job. For valid settings, see "JobFlagsEnum Constants" on page 41. The default is nqJobNoFlags.

Argument	Description
maxRunTimeMS	Optional. A long value that specifies the maximum time, in milliseconds, a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	Optional. A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	Optional. A date value that specifies the time the job is deactivated.

ScheduleJobNow Method

Schedules a new job with a Run Now trigger.

Syntax

```
ScheduleJobNow name, description, scriptType, script [, parameters,
flags, maxRunTimeMS]
```

Arguments

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) associated with the job.
parameters	Optional. A string array of parameter values passed to the script. The default is an empty array.
flags	Optional. A long value that specifies the flags associated with the job. For valid settings, see "JobFlagsEnum Constants" on page 41. The default is nqJobNoFlags.
maxRunTimeMS	Optional. A long value that specifies the maximum time, in milliseconds, a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.

ScheduleJobOnce Method

Schedules a new job with a Run Once trigger.

Syntax

```
ScheduleJobOnce name, description, scriptType, script, startDate,
startTime [, parameters, flags, maxRunTimeMS]
```

Arguments

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) associated with the job.
startDate	A date value that specifies the date the job is activated.
startTime	A date value that specifies the time the job is activated.
parameters	Optional. A string array of parameter values passed to the script. The default is an empty array.
flags	Optional. A long value that specifies the flags associated with the job. For valid settings, see "JobFlagsEnum Constants" on page 41. The default is nqJobNoFlag.
maxRunTimeMS	Optional. A long value that specifies the maximum time, in milliseconds, a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.

ScheduleJobWeekly Method

Schedules a new job with a Weekly trigger.

Syntax

```
ScheduleJobWeekly name, description, scriptType, script, startDate,
startTime, endTime, minutesInterval, weeksInterval, whichDays [,
parameters, flags, maxRunTimeMS, maxConcurrentInstances, endDate]
```

Argument

Argument	Description
name	A string that specifies the name of the job.
description	A string that specifies the description of the job.
scriptType	A string that specifies the script type associated with the job (either VBScript or JScript).
script	A string that specifies the script code or path (if the nqJobScriptContainsPath flag is set) associated with the job.
startDate	A date value that specifies the date the job is activated.
startTime	A date value that specifies the time the job is activated.
endTime	A date value that specifies the time the job is deactivated.
minutesInterval	A long value that specifies the number of minutes between consecutive job executions.
weeksInterval	An integer value that specifies the number of weeks between job invocations.
whichDays	An integer value that specifies the days of week the job runs. See DayOfWeekEnum values for valid settings.
parameters	Optional. A string array of parameter values passed to the script. The default is an empty array.
flags	Optional. A long value that specifies the flags associated with the job. For valid settings, see "JobFlagsEnum Constants" on page 41. The default is nqJobNoFlags.
maxRunTimeMS	Optional. A long value that specifies the maximum time, in milliseconds, a job runs before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
maxConcurrentInstances	Optional. A long value that specifies the maximum number of concurrent running instances of this job. The default is 0 (zero), which means no limit.
endDate	Optional. A date value that specifies the time the job is deactivated.

OnError Event

Occurs when the script engine encounters a run-time error while executing the script. This is intended for cleanup purposes, but the creative use of try/catch blocks in JScript and proper Error Handling in VBScript are often superior alternatives to using this event.

Syntax `OnError`

Usage The script engine calls this procedure when it encounters a run-time error while executing the script. Define this procedure in your script if you want to perform some cleanup activities before the script terminates, such as deleting temporary files and releasing resources.

Examples Example in VBScript:

```
Public Sub OnError()
   LogFile.WriteLine “Encountered a runtime error in the script.”
LogFile.Close
End Sub
```

Example in JScript:

```
function OnError()
{
   LogFile.WriteLine(“Encountered a runtime error in the
   script.”);
LogFile.Close();
}
```

Index

T

U

V

W