



SIEBEL ANALYTICS SCHEDULER GUIDE

VERSION 7.5

JULY 2002

12-BD5L3V

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404
Copyright © 2002 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.

The full text search capabilities of Siebel eBusiness Applications include technology used under license from Fulcrum Technologies, Inc. and are the copyright of Fulcrum Technologies, Inc. and/or its licensors.

Siebel, the Siebel logo, TrickleSync, TSQ, Universal Agent, and other Siebel product 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.

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 eBusiness 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

Introduction

How This Guide Is Organized	6
Revision History	7

Chapter 1. Using Siebel Analytics Scheduler

Concepts and Terms	10
The Back-End Database	11
Schemas	11
Scheduler Log File	13
Installation, Upgrade and Configuration Overview	14
High-Level Installation and Configuration Procedure	14
Upgrading from Siebel Analytics Version 7.0.4	15
Configuring Siebel Analytics Scheduler	17
About the Job Manager	24
Toolbar Options	24
Setting Up Jobs	27
Setting Up and Modifying General Job Properties	28
Defining Job Actions	30
Setting Job Triggers	31
Modifying iBots	35
Working with Job Instances	37
Working with Job Scripts	38
Working with the Script Object	38
Working with the Siebel Object	55



Index

Introduction

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

Intended users of this guide include system 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 will be useful primarily to people whose titles or job descriptions match one of the following:

Database Administrators	Persons who administer the database system, including data loading, system monitoring, backup and recovery, space allocation and sizing, and user account management.
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 system, including installing, maintaining, and upgrading Siebel applications.

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

How This Guide Is Organized

This guide is organized as a single chapter, with sections that explain the functionality of Siebel Analytics Scheduler, how to configure Siebel Analytics Scheduler, and how to work with jobs and iBots.

Revision History

Siebel Analytics Scheduler Guide, Version 7.5

Using Siebel Analytics Scheduler

1

This chapter provides information about configuring and using Siebel Analytics Scheduler.

Siebel Analytics Scheduler is an extensible scheduling application that allows you to schedule 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

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, a user defines 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: You must have licensed Siebel Delivers in order to create iBots. 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 17](#).

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 Jobs” on page 27](#), [“Working with Job Instances” on page 37](#), and [“Working with Job Scripts” on page 38](#).

The Back-End Database

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

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 subdirectory 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_JOBS, S_NQ_INSTANCES, S_NQ_JOBPARAMETERS, and S_NQ_ERRMSGs. 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 version 7.5 of Siebel eBusiness Applications OLTP. Therefore, if you have version 7.5 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 before Release 7.5. Therefore, if you are upgrading from Release 7.0.4, you must create the version 7.5 Scheduler tables and migrate your data from the old tables to the new tables. For more information, see [“Upgrading from Siebel Analytics Version 7.0.4”](#).

Table 1 describes the information stored in each table.

Table 1. Schema Tables

Table	Description
S_NQ_JOBS	Stores information about scheduled jobs.
S_NQ_INSTANCES	Stores information about instances.
S_NQ_JOBPARAMETERS	Stores information about job parameters.
S_NQ_ERRMSGs	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, which allows 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 subdirectory 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

This section provides an overview of how to install, upgrade, and configure Siebel Analytics Scheduler.

The following topics are covered in this section:

- [“High-Level Installation and Configuration Procedure” on page 14](#)
- [“Upgrading from Siebel Analytics Version 7.0.4” on page 15](#)
- [“Configuring Siebel Analytics Scheduler” on page 17](#)

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

High-Level Installation and Configuration Procedure

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 “Customizing Registry Settings Created During the Install,” in *Siebel Analytics Web Administration Guide*.

Upgrading from Siebel Analytics Version 7.0.4

The schema tables required to run Siebel Analytics Scheduler version 7.5 have changed from those of Siebel Analytics version 7.0.4. Therefore, to upgrade from Siebel Analytics version 7.0.4 to version 7.5, you need to create new Scheduler tables and migrate your data from those of Siebel Analytics version 7.0.4 into the new tables. For information about creating the Scheduler tables, see *Siebel Analytics Installation and Configuration Guide*. For information about migrating your data, see [“Migrating Data from Version 7.0.4 Tables to Version 7.5 Tables” on page 16](#).

If you are using Siebel Analytics as a stand-alone product, that is, without Siebel eBusiness Applications, you can continue to use the version 7.0.4 schema by running a registry script. For information on running the registry script, see [“Running a Registry Script To Maintain Version 7.0.4 Schema” on page 17](#).

Table 2 shows the tables names for version 7.0.4 and the corresponding new names for version 7.5.

Table 2. 7.0.4 and 7.5 Table Names

Version 7.0.4 Table Name	Version 7.5 Table Name
NQSJOBS	S_NQ_JOBS
NQSINSTANCES	S_NQ_INSTANCES
NQSJOBPARAMETERS	S_NQ_JOBPARAMETERS
NQSERRMSGs	S_NQ_ERRMSGs

Migrating Data from Version 7.0.4 Tables to Version 7.5 Tables

Use this procedure to migrate data from version 7.0.4 Scheduler tables to version 7.5 tables.

To migrate data

- 1 Locate the Schema subdirectory in the Siebel Analytics software installation directory.
- 2 Execute the script `migrate.SAACCT.xxx.sql` where `xxx` is your database type.
- 3 Execute the script `migrate.SAJOBS.xxx.sql` where `xxx` is your database type.

NOTE: The data migration will be successful only if the version 7.5 tables are in the same database as the version 7.0.4 tables.

Running a Registry Script To Maintain Version 7.0.4 Schema

Use this procedure if you are using Siebel Analytics as a standalone product and want to continue to use the version 7.0.4 schema.

CAUTION: The registry script creates entries in the registry telling the Scheduler to use the version 7.0.4 tables. After you run the registry script, the Scheduler will always look for the version 7.0.4 table names. If you later want to upgrade to the version 7.5 table names, you will need to manually delete the entries in the registry. To do so, delete the DB Column Names entry under `nQuire Software\nQuire Scheduler\7.0.`

To run the registry script

- 1 Locate the Schema subdirectory in the Siebel Analytics software installation directory.
- 2 If you are running IBM AS/400, execute `Use.7.0.OS390.AS400.reg`, otherwise execute `Use.7.0.SchTables.reg`.

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

- 1 Open the Siebel Analytics Administration Tool.
- 2 Select **Manage > Jobs** from the toolbar.

The Job Manager window appears.

- 3 Select 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 the Defaults button. To revert to the most recently used settings, click the Revert button.

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

Table 3. Scheduler Tab Fields

Field	Description
Database Type	The database type of the Scheduler back-end database. Select a type from the drop-down list.
Call Interface	The type of call the connection to the database will make. The call you select 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 will vary 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 will fail.
User name	<p>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.</p> <p>Note: It is recommended that the user name be the same as the table owner for databases like DB2 that require fully qualified table names (for example, SIEBEL.S_NQ_JOBS). Otherwise, Scheduler will not be set up properly.</p>

Table 3. Scheduler Tab Fields

Field	Description
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.
Timeout (Minutes)	The timeout specifies the amount of time, in minutes, that a connection to the data source will remain open after an operation completes. During this time, new operations will 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.
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.
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 will have a .scs extension. In general, users should not add or remove scripts from this directory. By default, this field is set to \$(NQUIREDIR)\Scripts\Scheduler where \$(NQUIREDIR) is the location in which the Siebel Analytics software is installed.
Default Script Path	This field refers to a path where user-created job scripts 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 will examine 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 \$(NQUIREDIR)\Scripts\Common where \$(NQUIREDIR) 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.
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.

Table 3. Scheduler Tab Fields

Field	Description
Purge Old instances after X days	Specifies the number of days after which old job instances will be deleted from the back-end database automatically. The default value is to delete after 7 days. To prevent old job instances from being deleted automatically, set the value to 0 (zero).
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.
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.
Maximum Execution Threads	This field specifies the maximum number of threads that are used in Siebel Analytics Scheduler's thread pool.
Pause When Service Starts	This is a flag that specifies that no jobs should execute when Siebel Analytics Scheduler starts. While it is paused, a user can add, modify, and remove jobs. However, no jobs will be executed. 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. Selecting Service Management > Continue Scheduling in the Job Manager will allow the Scheduler to continue with its regular execution.

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 the Defaults button. To revert to the most recently used settings, click the Revert button.

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

Table 4. Siebel Mail Tab Fields

Field	Description
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>.
Sender Address	This is the email address on the SMTP Server that will be 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 Server	This is the name of the SMTP server that will deliver the mail. The initial value is defaultmailserver.com, which must be changed to a valid server name in order to deliver mail.
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.
Number of Retries Upon Failure	If for any reason Siebel Delivers fails to deliver an email, it will try to resend it. The number specified in this field sets the number of retry attempts. The default is 1.
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.
Username	When the option Authenticate against SMTP Server is selected, this field specifies the user name for an email user on the specified SMTP server. If left blank, the email server will not authenticate any email from Siebel Delivers.
Password/Confirm Password	When the option Authenticate against SMTP Server is selected, these fields specify the user's password for the specified user name.

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 the Defaults button. To revert to the most recently used settings, click the Revert button.

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

Table 5. iBots Tab, Configuration Fields

Field	Description
Web Server	Specifies the URL of the server where Siebel Analytics Web resides. The default address is <code>http://localhost</code> . The mapped virtual directory must be <code>analytics</code> on this server, because this value will automatically be appended to the address. For example, the default address will be <code>http://localhost/analytics</code> at run time.
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 will be created. 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.
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 Administrator in this field. You must specify the actual Siebel Analytics Server Siebel "Administrator" username, not the username of a user with Siebel Analytics Server Administrator privileges.
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 <code>schconfig.exe</code> (UNIX).
Number of Global Tries	An iBot first tries to connect to the Web server to determine what information it is to deliver and to whom. This field specifies the number of times the Scheduler will attempt to connect to the server to get this information.

Table 5. iBots Tab, Configuration Fields

Field	Description
Minimum/ Maximum Global Sleep Seconds	The minimum and maximum number of seconds during which the iBot will randomly sleep before it attempts to reconnect to the server to get global information.
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 will attempt to connect to the server to issue these requests.
Minimum/ Maximum Request Sleep Seconds	The minimum and maximum number of seconds during which the iBot will randomly sleep before it attempts to reconnect to the server 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 will attempt to connect to the server to deliver the results.
Minimum/ Maximum Delivery Sleep Seconds	The minimum and maximum number of seconds during which the iBot will randomly sleep before it attempts to reconnect to the server to deliver results.
Max Concurrent Subscribers	Specifies how many concurrent actions or unique requests will be executed in parallel. This number represents the number of dedicated threads from the thread pool an iBot will use. You can override this setting for individual iBots in the Modify Job dialog box.

About the Job Manager

The Job Manager feature of Siebel Analytics Administration Tool allows you to schedule jobs to run at any time. 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, refer to the online help for Siebel Delivers.

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

Toolbar Options

This section describes the options available in the various menus on the Job Manager toolbar. [Table 6](#) provides a description of the File menu options.

Table 6. 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 <code>local</code> .
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 will be closed.

Table 7 provides a description of the Service Management menu options.

Table 7. 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 will continue 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	Allows the Scheduler to continue with its regular execution.
Stop Service	Stops the Scheduler service.

Table 8 provides a description of the Jobs menu options.

Table 8. 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. Once 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 9](#) provides a description of the Instances menu options.

Table 9. Instances Menu

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

Setting Up 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: The Job Manager does not allow you to add jobs that are iBots. iBots are passed into the Scheduler from the Siebel Analytics Server.

You can, however, modify iBots using the Modify Job dialog box. For more information, see [“Modifying iBots” on page 35](#).

You 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, select 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 Job Properties” on page 28](#)
- [“Defining Job Actions” on page 30](#)
- [“Setting Job Triggers” on page 31](#)

To modify a job

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

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

Setting Up and Modifying General 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 10](#) provides a description of the general job properties available in the Add Job and Modify Job dialog boxes.

Table 10. General 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 will describe 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 will 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 will execute 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 cancelled forcibly. If a job exceeds its runtime, it will fail 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 will execute. 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 10. General Job Properties

Field	Description
Delete Job When Done	When this option is selected, the job will be deleted 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 selected, since deleting a job can be done manually through the Job Manager.
Disabled	When this option is selected, the job script will not execute when its trigger expires. However, the next run time will still be updated according to the trigger settings. This is useful when testing or debugging a new job because it allows a Siebel administrator to quickly disable a job without losing all information.
Execute When Missed	When this option is selected, and Siebel Analytics Scheduler is stopped (either all scheduling is paused or the Scheduler application is stopped), the job will run when Siebel Analytics Scheduler is brought back up if its next run time was missed. When not selected, the job will execute at its next run time defined by its trigger.
Delete Script When Job is Removed	If this option is selected, 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 is used to disable the NQS Connect function in the job script. If a job script is not trusted by the Siebel administrator, this option can be selected 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 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 will perform when it executes. [Table 11](#) provides a description of the job action properties available in the Add Job and Modify Job dialog boxes.

Table 11. 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 selected.
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	<div>The values entered in this field are passed to the job script through the Parameters array. Enter one parameter per line. For example:</div> <pre>c:\siebel\data\scheduler cli_snowflake SELECT Lastname FROM Employee</pre> <div>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 <i>Siebel Analytics Installation and Configuration Guide</i>.</div>

Setting Job Triggers

A job trigger determines when and how often it will execute. Use the following fields in the Trigger area of the Add Job or Modify Job dialog box to define the actions a job will perform 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 select the trigger type. There are two single-run triggers, Run Now and Run Once.

Run Now

This trigger specifies that the job will be run immediately. It will execute only once.

Run Once

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

Recurrent Triggers

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

The fields described in [Table 12](#) pertain to all recurrent triggers.

Table 12. Recurrent Trigger Fields

Field	Description
Begin Date	This specifies the date when the first recurrent interval will be run. 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 selected.
End Date	This field becomes active when the option Has End Date is set. This specifies the date when the last recurrent interval will be run. If no end date is set, the job will run “forever.”

Table 12. Recurrent Trigger Fields

Field	Description
Start Time	This specifies the lower bounds of the recurrent interval. The job's first execution for a day will occur 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 will occur 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 will start its execution on the date specified in Begin Date at 11:00 PM and will continue until 2:00 AM on the following day.
Has End Date	If this option is not selected, the job will run in perpetuity. If this option is selected, specify an End Date.
Set Start Time To Now	If this option is selected, 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 will start execution promptly at its Start Time, and will execute 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), it is possible that a long-running job could have overlapping executions. Use this field to set the number of concurrent running instances that will be allowed. To allow any number of concurrent instances, set this value to zero.

Recurrent Trigger Types

The fields described in [Table 13](#) are the recurrent triggers available from the Trigger Type drop-down list. Depending on the trigger type you select, additional options will become active. Refer to the examples to see how these additional options can be used.

Table 13. Recurrent Trigger Types

Type	Description
Daily	<p>Triggers of this type allow a job to be run every few days. The Days Interval field specifies the number of days between each subsequent recurrent interval.</p> <p>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.</p> <p>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.</p>
Weekly	<p>Triggers of this type allow a job to be run on certain 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 will occur.</p> <p>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.</p>

Table 13. Recurrent Trigger Types

Type	Description
Monthly by Date	<p>Triggers of this type allow a job to recur on specific days of the month. The Months field specifies which months this job will execute. The Days field specifies which days of those months. If the given day does not exist for a given month, that day is ignored.</p> <p>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.</p> <p>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.</p>
Monthly by DOW	<p>Triggers of this type allow a job to run on specific occurrences of certain days of the week during given months. The Months field specifies which months this job will execute. The Days of the Week field specifies which days of the week the job will execute 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 will be used, depending on whether there are four or five occurrences during that month.</p> <p>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.</p> <p>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.</p>

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 select Jobs > Modify Job.

The Modify Job dialog box displays the iBot you selected. Modify the iBot properties, as described in [Table 14](#).

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 14. iBot Properties

Web Server	Specifies the Web server this iBot will contact when it runs. Note: It is recommended that you do not change this setting, because the iBot may not exist on a different Web server. This feature was added for debugging purposes only.
-------------------	---

Table 14. iBot Properties

Debug Log	Determines whether debugging information is written to a log.
Max Concurrent Subscribers	<p>Specifies how many concurrent actions or unique requests will be executed in parallel.</p> <p>Note: This number represents the number of dedicated threads from the thread pool that this iBot will use. 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 will be able to handle more subscribers concurrently. However, you should exercise caution, because if a subscriber receives multiple deliveries, threads can become blocked.</p> <p>For example, if you have two subscribers, each with a delivery to an email device and to the Intelligence Dashboard, then two threads will be able to 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 will try to execute, but two will occupy threads that could be used by other iBots. The other iBots will not be able to execute if one long-running iBot uses all of the threads.</p> <p>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.</p>

Working with 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 will be issued to Siebel Analytics Scheduler and the instance will be marked as cancelled when its registered cancel methods are called. For more information about registered cancel methods, see the command [“RegisterCancelCommand” on page 43](#).
- 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 the Delete key.
 - 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 select the End Time method, all jobs with an End Time less than or equal to the given time are purged. Select 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.

Working with 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.

Working with the Script Object

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.

Read-Only Script Object Properties

This section describes the read-only script object properties.

Script

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” on page 43](#).

Return Value: returns a script object.

JobID

Returns the job identification number associated with this instance.

Return Value: returns a long value.

InstanceID

Returns the instance identification number associated with this instance.

Return Value: returns a double value.

UserID

Returns the user identification number associated with the instance.

Return Value: returns a string value.

Parameter Count

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

Return Value: returns a long value.

Parameter (index)

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)
```

where:

`index` is the zero-based index of the parameter.

Return Value: returns a string value.

Read/Write Script Object Properties

This section describes the read/write script object properties.

Error Message

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 will be 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 will have a Failed status and its ErrorMessage property will be set. Resetting this value to the empty string (“”) will clear the error messages.

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

ExitCode

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

This section describes the script-defined constants. These constants are used by the methods that allow a script to schedule new jobs.

JobFlagsEnum

The JobFlagsEnum values are used with the scheduling methods of the Script object to control how a job behaves. [Table 15](#) describes JobFlagsEnum values.

Table 15. JobFlagsEnum alues

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.

Table 15. JobFlagsEnum alues

Constant	Description
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 will be 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.

DayOfWeekEnum

The DayOfWeekEnum values are used with the scheduling functions to identify days in a week. [Table 16](#) describes DayOfWeekEnum values.

Table 16. DayOfWeekEnum Values

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

MonthEnum

The MonthEnum values are used with the scheduling functions to identify months. [Table 17](#) describes MonthEnum values.

Table 17. 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

The OccurrenceEnum values are used with the scheduling functions to identify the occurrence of a given day. [Table 18](#) describes OccurrenceEnum values.

Table 18. OccurrenceEnum Values

Constant	Description
nqFirst	First occurrence
nqSecond	Second occurrence
nqThird	Third occurrence

Table 18. OccurrenceEnum Values

Constant	Description
nqFourth	Fourth occurrence
nqLast	Last occurrence

DayEnum

The DayEnum values are used with the scheduling functions to identify days in a month, from Day 1 to Day 31. [Table 19](#) describes DayEnum values.

Table 19. DayEnum Values

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

Script Object Methods

This section describes the script object methods.

RegisterCancelCommand

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

Syntax:

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

where:

`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 Notes: Occasionally, an object's method takes a long time to complete. If the job is cancelled 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 allowing the registration of a method that is asynchronously called by the script engine if the script gets cancelled.

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

Since it is often good practice to hide implementation details of a COM object from the caller, consider having the COM object itself handle all registration and deregistration of cancel commands. A COM object should allow the script writer to pass it an instance of the Script object to the COM object. The COM object could then call the `RegisterCancelCommand()` and `DeregisterCancelCommand()` methods because the Script object implements the `IDispatch` interface.

DeregisterCancelCommand

Deregisters a previously registered cancel method.

Syntax:

```
DeregisterCancelCommand
```

Usage Notes: 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 cancelled.

NQSConnect

Opens a connection to a Siebel data source and returns a Siebel object.

Syntax:

```
Set nqs = NQSConnect ( dsn [, timeout] )
```

where:

<code>dsn</code>	A String that specifies the data source name to connect to.
<code>timeout</code>	Optional. A Long value that specifies the number of seconds to wait for the connection to open. The default is 30. Specifying 0 (zero) will wait indefinitely.

Return Value: Returns a Siebel object.

Usage Notes: Call this method to connect to a Siebel data source. Once connected, you can use the Siebel object's properties and methods to send queries and retrieve records. See [“Working with the Siebel Object” on page 55](#) for more information.

Please note that this method is *not* available for jobs that are not trusted (that is, the `nqJobIsUserScript` flag is set).

ScheduleJobNow

Schedules a new job with a Run Now trigger.

Syntax:

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

where:

<code>name</code>	A string that specifies the name of the job.
<code>description</code>	A string that specifies the description of the job.
<code>scriptType</code>	A string that specifies the script type associated with the job (either VBScript or JScript).
<code>script</code>	A string that specifies the script code or path (if the <code>nqJobScriptContainsPath</code> flag is set) associated with the job.
<code>parameters</code>	Optional. A string array of parameter values passed to the script. The default is an empty array.
<code>flags</code>	Optional. A long value that specifies the flags associated with the job. For valid settings, see “JobFlagsEnum” on page 40 . The default is <code>nqJobNoFlags</code> .
<code>maxRunTimeMS</code>	Optional. A long value that specifies the maximum time, in milliseconds, a job is allowed to run before it is terminated. The default is 0 (zero), which means the job can run indefinitely.

ScheduleJobOnce

Schedules a new job with a Run Once trigger.

Syntax:

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

where:

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” on page 40 . The default is nqJobNoFlag.
maxRunTimeMS	Optional. A long value that specifies the maximum time, in milliseconds, a job is allowed to run before it is terminated. The default is 0 (zero), which means the job can run indefinitely.

ScheduleJobDaily

Schedules a new job with a Daily trigger.

Syntax:

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

where:

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).

<code>script</code>	A String that specifies the script code or path (if the <code>nqJobScriptContainsPath</code> flag is set) associated with the job.
<code>startDate</code>	A date value that specifies the date the job is activated.
<code>startTime</code>	A date value that specifies the time the job is activated.
<code>endTime</code>	A date value that specifies the time the job is deactivated.
<code>minutesInterval</code>	A long value that specifies the number of minutes between consecutive job executions.
<code>daysInterval</code>	An integer value that specifies the number of days between job invocations.
<code>parameters</code>	Optional. A string array of parameter values passed to the script. The default is an empty array.
<code>flags</code>	Optional. A long value that specifies the flags associated with the job. For valid settings, see “JobFlagsEnum” on page 40 . The default is <code>nqJobNoFlags</code> .
<code>maxRunTimeMS</code>	Optional. A long value that specifies the maximum time, in milliseconds, a job is allowed to run before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
<code>maxConcurrentInstances</code>	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.
<code>endDate</code>	Optional. A date value that specifies the time the job is deactivated.

ScheduleJobWeekly

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]
```

where:

<code>name</code>	A string that specifies the name of the job.
-------------------	--

<code>description</code>	A string that specifies the description of the job.
<code>scriptType</code>	A string that specifies the script type associated with the job (either VBScript or JScript).
<code>script</code>	A string that specifies the script code or path (if the <code>nqJobScriptContainsPath</code> flag is set) associated with the job.
<code>startDate</code>	A date value that specifies the date the job is activated.
<code>startTime</code>	A date value that specifies the time the job is activated.
<code>endTime</code>	A date value that specifies the time the job is deactivated.
<code>minutesInterval</code>	A long value that specifies the number of minutes between consecutive job executions.
<code>weeksInterval</code>	An integer value that specifies the number of weeks between job invocations.
<code>whichDays</code>	An integer value that specifies the days of week the job runs. See <code>DayOfWeekEnum</code> values for valid settings.
<code>parameters</code>	Optional. A string array of parameter values passed to the script. The default is an empty array.
<code>flags</code>	Optional. A long value that specifies the flags associated with the job. For valid settings, see “JobFlagsEnum” on page 40 . The default is <code>nqJobNoFlags</code> .
<code>maxRunTimeMS</code>	Optional. A long value that specifies the maximum time, in milliseconds, a job is allowed to run before it is terminated. The default is 0 (zero), which means the job can run indefinitely.
<code>maxConcurrentInstances</code>	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.
<code>endDate</code>	Optional. A date value that specifies the time the job is deactivated.

ScheduleJobMonthlyDate

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]
```

where:

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” on page 43 .
whichMonths	An integer value that specifies the months the job runs. For valid settings, see “MonthEnum” 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” on page 40 . The default is nqJobNoFlags.
maxRunTimeMS	Optional. A long value that specifies the maximum time, in milliseconds, a job is allowed to run 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

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]
```

where:

<code>name</code>	A string that specifies the name of the job.
<code>description</code>	A string that specifies the description of the job.
<code>scriptType</code>	A string that specifies the script type associated with the job (either VBScript or JScript).
<code>script</code>	A string that specifies the script code or path (if the <code>nqJobScriptContainsPath</code> flag is set) associated with the job.
<code>startDate</code>	A date value that specifies the date the job activated.
<code>startTime</code>	A date value that specifies the time the job is activated.
<code>endTime</code>	A date value that specifies the time the job is deactivated.
<code>minutesInterval</code>	A long value that specifies the number of minutes between consecutive job executions.
<code>whichOccurrences</code>	An integer value that specifies the occurrences of days of week the job runs. For valid settings, see “DayEnum” on page 43 .
<code>whichDays</code>	An integer value that specifies the days of week the job runs. For valid settings, see “DayOfWeekEnum” on page 41 .
<code>whichMonths</code>	An integer value that specifies the months the job runs. For valid settings, see “MonthEnum” on page 42 .
<code>parameters</code>	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” on page 40 . The default is nqJobNoFlags.
maxRunTimeMS	Optional. A long value that specifies the maximum time, in milliseconds, a job is allowed to run 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.

CreateArray

Creates an Array object.

Syntax:

Syntax 1	Set array = CreateArray ()
Syntax 2	Set array = CreateArray (size)
Syntax 3 elementn)	Set array = CreateArray (element0, element1, ..., elementn)

where:

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 <i>n</i> , respectively.

Return Value: returns an Array object.

Usage Notes: 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;
}
```

LaunchProcess

Executes a command line in a new process.

Syntax:

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

where:

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 Notes: 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.

GetTempFileName

Returns a temporary file name.

Syntax:

```
tfname = GetTempFileName()
```

Return Value: returns a string value.

Usage Notes: `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.

GetConfigurationValue

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

Syntax:

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

where:

`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 Notes: `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.

Script Object Events

This section describes the script object events.

OnError

Occurs when the script engine encounters a runtime 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 Notes: The script engine calls this procedure when it encounters a runtime 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.

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();  
}
```

Working with the Siebel Object

A Siebel object represents an active connection to a Siebel Analytics Server. This is the object returned by the `NQSCONNECT()` method of the Script object.

Read-Only Siebel Object Properties

This section describes the read-only Siebel object properties.

Query

Returns the query string used to select records for the record set of a Siebel object. This is not a valid string until after issuing a call to `OpenRecordset()`.

Return Value: returns a string value.

Empty

Returns True if there are no records in the record set of a Siebel object.

Return Value: returns a boolean value.

EOF

Returns True if the record set cursor is positioned after the last record.

Return Value: returns a boolean value.

FieldCount

Returns the number of fields in the record set.

Return Value: returns an Integer value.

FieldName (index)

Returns the name of a specific field in the record set of a Siebel object.

where:

`index` The zero-based index of the field.

Return Value: returns a string value.

Field(index)

Returns the value of a specific field in the record set of a Siebel object.

Syntax:

Syntax 1 Field (index)

Syntax 2 Field (fieldName)

where:

index The zero-based index of the field.

fieldName The name of the field.

Return Value: returns a variant value.

Read/Write Siebel Object Properties

This section describes the read/write Siebel object properties.

MaxRows

Sets or returns the maximum rows to retrieve.

Settings and return value: sets or returns a long value. The default value of 0 (zero) specifies that any number of rows is acceptable.

CacheSize

Sets or returns the maximum rows to cache in memory for a query operation.

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

Siebel Object Methods

This section describes the Siebel object methods.

OpenRecordset

Opens a record set by executing the given query.

Syntax:

OpenRecordset query [, timeout]

where:

query A string that specifies the SQL statement to execute.

timeout Optional. A long value that specifies the number of seconds to wait for the query execution to complete. The default is 30. Specifying 0 (zero) will cause the call to block until a result set exists.

Usage Notes: Call this method to send a query and retrieve results from a Siebel data source.

CloseRecordset

Closes the current record set.

Syntax:

```
CloseRecordset
```

Usage Notes: Call this method to close a previously opened record set.

Next

Moves the cursor to the next record in a record set.

Syntax:

```
Next
```

Usage Notes: Call this method to move the cursor to the next record. If the cursor is moved past the last record, the EOF method will return True.

Example of NQSConnect Object

This section shows an example of an NQSConnect object.

```
/*****
```

```
File:
```

```
    TestNQSConnect.js
```

```
Description:
```

```
    This is a job script that connects to an NQS
    server, executes a query, and dumps the result set
    to a file.
```

```
Parameters:
```

```
Parameter(0) specifies the output file Path.
Parameter(1) specifies a DSN to a Siebel server.
Parameter(2) specifies the query to execute.
```

```

*****/

/** Create Output File **/
var fso = new ActiveXObject("Scripting.FileSystemObject");
var filename = Parameter(0) + "\\\" + "output.\" + JobID + \".\" +
    InstanceID + \".txt";
var a = fso.CreateTextFile(filename, true);

/** Dump Job Attributes **/
a.WriteLine("Script: TestNQSConnect.js");
a.WriteLine("JobID: " + JobID.toString());
a.WriteLine("InstanceID: " + InstanceID.toString());
a.WriteLine("UserID: " + UserID);
a.WriteLine("ParameterCount: " + ParameterCount.toString());

/** Connect to Server **/
a.Write("Connecting to " + Parameter(1) + " ... ");
var nqs = NQSConnect(Parameter(1));
a.WriteLine("OK");

/** Execute Query **/
a.Write("Executing query \" + Parameter(2) + "\" ... ");
nqs.OpenRecordset(Parameter(2), 0);
a.WriteLine("OK");

/** Dump Column Names **/
var sep = "\t";
var i;
for (i=0; i<nqs.FieldCount; ++i)
{
    a.Write(nqs.FieldName(i) + sep);
}
a.WriteLine();

/** Dump Result Set **/
while (!nqs.EOF)
{
    for (i=0; i<nqs.FieldCount; i++)
    {
        a.Write(String(nqs.Field(i)) + sep);
    }
    a.WriteLine();
    nqs.Next();
}

/** Close Result Set **/
```

```
nqs.CloseRecordset();

/** Close Output File */
a.Close();

function OnError()
{
    a.Close();
}
```


Index

A

audience description 5
Authenticate against SMTP Server
 configuration field 21

B

Bulk Fetch Buffer Size (bytes) configuration
 field 19

C

CacheSize property 56
Call Interface configuration field 18
canceling instances 26
CloseRecordset method 57
COM object, configuration fields 20
concurrent connections, maximum
 number 19
CreateArray object method 51

D

daily trigger type 33
Data Source Name configuration field 18
data types, about changing 12
Database Type configuration field 18
database, back end
 database type 18
database, back-end
 bulk fetch buffer size 19
 call interface type 18
 configuration options 18
 data source logon name 18
 default configuration options,
 restoring 18
 job script path 19

logon username 18
maximum concurrent connections
 field 19
password field 19
purging 20
Siebel Analytics Scheduler
 requirement 12
 timeout specification 19
databases supported 11
day-of-month flags 43
day-of-week constants 41
DayOfWeek Enum values 41
Default Script Path configuration field 19
Delete Job When Done field 29
Delete Script When Job is Removed 29
DeregisterCancelCommand object
 method 44
Disable NQS Functions field 29
Disabled field 29

E

Empty property 55
EOF property 55
Error Message script object property 39
Execute When Missed field 29
ExitCode script object property 40

F

Field(index) property 55
FieldCount property 55
FieldName (index) property 55
From Display Name configuration field 21

G

GetTempFileName object method 53

I

- iBot Log Directory configuration field 22
- InstanceID script object property 38
- instances
 - canceling, purging, or viewing 26
 - definition of 10
 - job instances, managing 37

J

- job actions, defining 30
- job instances, managing 37
- Job Manager
 - File menu options, table of 24
 - Instances menu options, table of 26
 - job instances, managing 37
 - Jobs menu options, table of 25
 - overview 24
 - Service Management menu options, table of 25
 - shutting down 24
 - Siebel iBot jobs, adding and modifying 27
- Job Manager Configuration dialog box
 - accessing 17
 - opening 24
 - role of 17
 - Scheduler Tab fields 18
 - Siebel iBots tab fields 22
 - Siebel Mail tab fields 20
- job parameters, definition of 10
- job script path 19
- job scripts
 - day-of-month flags 43
 - day-of-week flags 41
 - definition of 10
 - deleting at end of job 29
 - disabling 29
 - disabling connect function 29
 - locating 12
 - month-of-year constants 42
 - occurrence constants 42
 - Script object 38

- Script object read/write properties 39
- Script object read-only properties 38
- Script-defined constants 40

job triggers

- recurrent trigger fields 31
- recurrent trigger types 33
- setting 31
- single-run triggers 31

- JobFlagsEnum flag values 40

- JobID field 28

- JobID script object property 38

jobs

- about setting up 27
- adding or modifying jobs 27
- automatic deletion after execution 29
- definition of 10
- job property configuration fields 28
- last run time and next run time fields 28
- number of instances running 28
- scheduling and canceling 25
- time out setting 28

L

- Last Run Time field 28
- LaunchProcess object method 52
- log file 13
- login username, back-end database 18

M

- managing job instances 37
- Max Concurrent Subscribers configuration field 23
- Maximum Connections configuration field 19
- Maximum Execution Threads configuration field 20
- Maximum Run Time MS field 28
- MaxRows property 56
- Minimum Execution Threads configuration field 20
- Minimum/Maximum Delivery Sleep Seconds configuration field 23

Minimum/Maximum Global Sleep Seconds
configuration field 23

Minimum/Maximum Request Sleep
Seconds configuration field 23

Monthly by Date trigger type 34

Monthly by DOW trigger type 34

month-of-year constants 42

N

Next method 57

nqJobDeleteScriptWhenDone script-defined
constant 40

nqJobDeleteWhenDone script-defined
constant 40

nqJobDisabled script-defined constant 40

nqJobExecuteWhenMissed script-defined
constant 40

nqJobHasEndDate script-defined
constant 40

nqJobIsUserScript script-defined
constant 41

nqJobNoFlags 40

nqJobScriptContainsPath script-defined
constant 40

nqJobStartNow script-defined constant 41

nQS (Siebel) Administrator Name
configuration field 22

nQS (Siebel) Administrator Password/
Confirm Password configuration
field 22

NQScheduler.log 13

nQScheduler.log file 13

NQSCONNECT Object
example 57

Siebel Object 55

NQSCONNECT object method 44

Number of Delivery Tries configuration
field 23

Number of Global Tries configuration
field 22

Number of Request Tries configuration
field 23

Number of Retries Upon Failure
configuration field 21

O

occurrence constants, table of values 42

OnError object event 54

OpenRecordset method 56

P

Parameter (index) script object
property 39

ParameterCount script object property 39

Parameters (script) field 30

Password/Confirm Password configuration
field 19, 21

Pause When Service Starts configuration
field 20

pausing and continuing Siebel Analytics
Scheduler 25

Purge DB every X minutes configuration
field 20

Purge Old instances after X days
configuration field 20

purging instances 26

Q

Query property 55

R

recurrent triggers
fields related to 31

trigger types 33

RegisterCancelCommand object
method 43

Running Instance Count field 28

running job, definition of 10

S

S_NQ_ERRMSGs 11

S_NQ_INSTANCES 11

S_NQ_JOBPARAMETERS 11

S_NQ_JOBS 11

- schconfig.exe file 14
 - ScheduleJobDaily object method 46
 - ScheduleJobMonthlyDate object method 48
 - ScheduleJobMonthlyDOW object method 50
 - ScheduleJobNow object method 45
 - ScheduleJobOnce object method 45
 - ScheduleJobWeekly object method 47
 - Scheduler Log File 13
 - Scheduler Script Path configuration field 19
 - schema tables 11
 - Script Contains File Name field 30
 - Script field 30
 - Script object
 - overview 38
 - read/write properties 39
 - read-only properties 38
 - Script object property 38
 - Script Type field 30
 - script-defined constants
 - about 40
 - day-of-month flags 43
 - day-of-week constants 41
 - job status flags 40
 - month-of-year flags 42
 - occurrence flags 42
 - scripts
 - object events 54
 - object methods 43
 - Sender Address configuration field 21
 - Siebel Analytics Scheduler
 - COM default values, restoring 20
 - connection, opening and closing 24
 - databases supported 11
 - execution threads 20
 - Job Manager, role of 24
 - overview 9
 - pausing and continuing 25
 - pausing execution 20
 - role of 10
 - scheduling and removing jobs 25
 - schema tables 11
 - stopping 25
 - Siebel Analytics Scheduler configuration
 - configuration options 17
 - Job Manager Configuration dialog box, opening 24
 - overview 14
 - registry settings, configuring 15
 - Scheduler Tab options 18
 - Siebel Analytics Scheduler installation 14
 - Siebel iBots
 - debugging information 36
 - default values, restoring 22
 - Job Manager Configuration fields 22
 - jobs, adding and modifying 27
 - log file location 22
 - maximum concurrent users 23
 - maximum number of threads 36
 - modifying 35
 - number connection attempts 22
 - number delivery attempts 23
 - property fields 35
 - Siebel Analytics Scheduler, role of 9
 - sleep time 23
 - Web server 35
 - Siebel Object
 - methods 56
 - NQSCONNECT Object 55
 - read/write properties 56
 - read-only properties 55
 - Siebel OLTP schema and Siebel Analytics Scheduler schema tables 11
 - Single-Run Triggers 31
 - SMTP Port configuration field 21
 - SMTP Server configuration field 21
 - stopping Siebel Analytics Scheduler 25
- ## T
- Temporary File Path configuration field 19
 - temporary files, location of 19
 - Timeout (Minutes) configuration field 19
 - trigger, definition of 10

U

- User name configuration field 18
- UserID script object property 38
- username 18
- Username configuration field 21

V

- viewing instances 26

W

- Web Server configuration field 22
- weekly trigger type 33

