SYBASE®

Job Scheduler User's Guide

Adaptive Server® Enterprise

15.0

Contents

About This Book

Audience

This book describes how to create and schedule jobs in Sybase® Adaptive Server® Enterprise databases.

This book is intended for System Administrators and Database Administrators.

How to use this book

This book assists you in creating scheduled jobs. It contains these chapters:

- Chapter 1, "Overview," provides an overview of the components and architecture of Job Scheduler.

- Chapter 2, "Configuring and Running Job Scheduler," provides the necessary information to install, configure, and run Job Scheduler.

- Chapter 3, "Using Templates to Create Jobs," describes the use of templates with Job Scheduler.

- Chapter 4, "Using Job Scheduler at the Command Line," provides examples for creating and scheduling jobs from the command line.

- Chapter 5, "Command Reference," describes the properties and parameters of the command line stored procedures.

- Chapter 6, "Managing Job Scheduler in the Sybase Central ASE Plug-in," explains the procedures for Job Scheduler in the Sybase Central GUI.

- Chapter 7, "Troubleshooting," provides solutions for potential problems using Job Scheduler.

Related documents

The Adaptive Server documentation set consists of the following:

- The release bulletin for your platform – contains last-minute information that was too late to be included in the books.

  A more recent version of the release bulletin may be available on the World Wide Web. To check for critical product or document information that was added after the release of the product CD, use the Sybase Technical Library.

- The *Installation Guide* for your platform – describes installation, upgrade, and configuration procedures for all Adaptive Server and related Sybase products.

- *What's New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 15.0, the system changes added to support those features, and changes that may affect your existing applications.

- *ASE Replicator User's Guide* – describes how to use the Adaptive Server Replicator feature of Adaptive Server to implement basic replication from a primary server to one or more remote Adaptive Servers.

- *Component Integration Services User's Guide* – explains how to use the Adaptive Server Component Integration Services (CIS) feature to connect remote Sybase and non-Sybase databases.

- The *Configuration Guide* for your platform – provides instructions for performing specific configuration tasks for Adaptive Server.

- *Full-Text Search Specialty Data Store User's Guide* – describes how to use the Full-Text Search feature with Verity to search Adaptive Server Enterprise data.

- *Glossary* – defines technical terms used in the Adaptive Server documentation.

- *Historical Server User's Guide* – describes how to use Historical Server to obtain performance information for SQL Server® and Adaptive Server.

- *Java in Adaptive Server Enterprise* – describes how to install and use Java classes as datatypes, functions, and stored procedures in the Adaptive Server database.

- *Messaging Service User's Guide* – describes how to use Real Time Messaging Services to integrate TIBCO Java Message Service and IBM WebSphere MQ messaging services with all Adaptive Server database applications.

- *Monitor Client Library Programmer's Guide* – describes how to write Monitor Client Library applications that access Adaptive Server performance data.

- *Monitor Server User's Guide* – describes how to use Monitor Server to obtain performance statistics from SQL Server® and Adaptive Server.

- *Performance and Tuning Guide* – is a series of four books that explains how to tune Adaptive Server for maximum performance:

- *Basics* – provides the basics for understanding and investigating performance questions in Adaptive Server.

- *Locking* – describes how the various locking schemas can be used for improving performance in Adaptive Server.

- *Optimizer and Abstract Plans* – describes how the optimizer processes queries and how abstract plans can be used to change some of the optimizer plans.

- *Monitoring and Analyzing* – explains how statistics are obtained and used for monitoring and optimizing performance.

- *Quick Reference Guide* – provides a comprehensive listing of the names and syntax for commands, functions, system procedures, extended system procedures, datatypes, and utilities in a pocket-sized book.

- *Reference Manual* – is a series of four books that contains the following detailed Transact-SQL® information:

  - *Building Blocks* – Transact-SQL datatypes, functions, global variables, expressions, identifiers and wildcards, and reserved words.

  - *Commands* – Transact-SQL commands.

  - *Procedures* – Transact-SQL system procedures, catalog stored procedures (CSPs), system extended stored procedures, and dbcc stored procedures.

  - *Tables* – Transact-SQL system tables and dbcc tables.

- *System Administration Guide* – provides in-depth information about administering servers and databases. This manual includes instructions and guidelines for managing physical resources, security, user and system databases, and specifying character conversion, international language, and sort order settings.

- *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Available only in print version.

- *Transact-SQL User's Guide* – documents Transact-SQL, Sybase's enhanced version of the relational database language. This manual serves as a textbook for beginning users of the database management system. It also contains descriptions of the pubs2 and pubs3 sample databases.

- *Using Adaptive Server Distributed Transaction Management Features* – explains how to configure, use, and troubleshoot Adaptive Server DTM features in distributed transaction processing environments.

- *Using Sybase Failover in a High Availability System* – provides instructions for using Sybase's Failover to configure an Adaptive Server as a companion server in a high availability system.

- *Unified Agent and Agent Management Console* – Describes the Unified Agent, which provides runtime services to manage, monitor, and control distributed Sybase resources.

- *Utility Guide* – documents the Adaptive Server utility programs, such as isql and bcp, which are executed at the operating system level.

- *Web Services User's Guide* – explains how to configure, use, and troubleshoot Web Services for Adaptive Server.

- *XA Interface Integration Guide for CICS, Encina, and TUXEDO* – provides instructions for using the Sybase DTM XA interface with X/Open XA transaction managers.

- *XML Services in Adaptive Server Enterprise* – describes the Sybase native XML processor and the Sybase Java-based XML support, introduces XML in the database, and documents the query and mapping functions that comprise XML Services.

Other sources of information

Use the Sybase Getting Started CD, the SyBooks™ CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.

- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

  Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

  Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

   To access the Sybase Product Manuals Web site, go to Product Manuals at http://www.sybase.com/support/manuals/.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ **Finding the latest information on product certifications**

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.

2 Select Products from the navigation bar on the left.

3 Select a product name from the product list and click Go.

4 Select the Certification Report filter, specify a time frame, and click Go.

5 Click a Certification Report title to display the report.

❖ **Finding the latest information on component certifications**

1 Point your Web browser to Availability and Certification Reports at http://certification.sybase.com/.

2 Either select the product family and product under Search by Product; or select the platform and product under Search by Platform.

3 Select Search to display the availability and certification report for the selection.

❖ **Creating a personalized view of the Sybase Web site (including support pages)**

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.

2 Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ **Finding the latest information on EBFs and software maintenance**

1 Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.

2 Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.

3 Select a product.

4 Specify a time frame and click Go. A list of EBFs/Maintenance releases is displayed.

Padlock icons indicate that you do not have download authorization for certain EBFs/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the "Technical Support Contact" role to your MySybase profile.

5 Click the Info icon to display the EBFs/Maintenance report, or click the product description to download the software.

Conventions

The following sections describe conventions used in this manual.

SQL is a free-form language. There are no rules about the number of words you can put on a line or where you must break a line. However, for readability, all examples and most syntax statements in this manual are formatted so that each clause of a statement begins on a new line. Clauses that have more than one part extend to additional lines, which are indented. Complex commands are formatted using modified Backus Naur Form (BNF) notation.

Table 1 shows the conventions for syntax statements that appear in this manual:

Table 1: Font and syntax conventions for this manual

Element	Example
Command names, procedure names, utility names, and other keywords display in sans serif font.	select
	sp_configure
Database names and datatypes are in sans serif font.	master database
Book names, file names, variables, and path names are in italics.	*System Administration Guide*
	sql.ini file
	column_name
	$SYBASE/ASE directory

Element	Example

Variables—words that stand for values that you fill in—when they are part of a query or statement, are in italics in Courier font.	`select column_name`
	`    from table_name`
	`    where search_conditions`

Type parentheses as part of the command.	`compute row_aggregate (column_name)`
Double colon, with equals sign indicates that the syntax is written in BNF notation. Do not type this symbol. Indicates "is defined as".	`::=`
Curly braces mean that you must choose at least one of the enclosed options. Do not type the braces.	`{cash, check, credit}`
Brackets mean that to choose one or more of the enclosed options is optional. Do not type the brackets.	`[cash	check	credit]`
The comma means you may choose as many of the options shown as you want. Separate your choices with commas as part of the command.	`cash, check, credit`
The pipe or vertical bar (	) means you may select only one of the options shown.	`cash	check	credit`
An ellipsis (...) means that you can *repeat* the last unit as many times as you like.	`buy thing = price [cash	check |` `credit]` `[, thing = price [cash	check |` `credit]]...` You must buy at least one thing and give its price. You may choose a method of payment: one of the items enclosed in square brackets. You may also choose to buy additional things: as many of them as you like. For each thing you buy, give its name, its price, and (optionally) a method of payment.

- Syntax statements (displaying the syntax and all options for a command) appear as follows:

      sp_dropdevice [*device_name*]

  For a command with more options:

      select *column_name*
        from *table_name*
        where *search_conditions*

  In syntax statements, keywords (commands) are in normal font and identifiers are in lowercase. Italic font shows user-supplied words.

- Examples showing the use of Transact-SQL commands are printed like this:

      select * from publishers

- Examples of output from the computer appear as follows:

```
pub_id   pub_name               city         state
-------  ---------------------  -----------  -----
0736     New Age Books          Boston       MA
0877     Binnet & Hardley       Washington   DC
1389     Algodata Infosystems   Berkeley     CA
```

```
(3 rows affected)
```

In this manual, most of the examples are in lowercase. However, you can disregard case when typing Transact-SQL keywords. For example, SELECT, Select, and select are the same.

Adaptive Server's sensitivity to the case of database objects, such as table names, depends on the sort order installed on Adaptive Server. You can change case sensitivity for single-byte character sets by reconfiguring the Adaptive Server sort order. For more information, see the *System Administration Guide*.

Accessibility features

This document is available in an HTML version that is specialized for accessibility. You can navigate the HTML with an adaptive technology such as a screen reader, or view it with a screen enlarger.

Job Scheduler and the HTML documentation have been tested for compliance with U.S. government Section 508 Accessibility requirements. Documents that comply with Section 508 generally also meet non-U.S. accessibility guidelines, such as the World Wide Web Consortium (W3C) guidelines for Web sites.

The online help for this product is also provided in HTML, which you can navigate using a screen reader.

Note You might need to configure your accessibility tool for optimal use. Some screen readers pronounce text based on its case; for example, they pronounce ALL UPPERCASE TEXT as initials, and Mixed Case Text as words. You might find it helpful to configure your tool to announce syntax conventions. Consult the documentation for your tool.

For information about how Sybase supports accessibility, see Sybase Accessibility at http://www.sybase.com/accessibility. The Sybase Accessibility site includes links to information on Section 508 and W3C standards.

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

CHAPTER 1   **Overview**

This chapter provides an overview of the components and architecture of Job Scheduler.

Topic	Page
Introduction	1
Terminology and concepts	2
Job Scheduler components and functionality	3
Job Scheduler architecture	3
Security	6

Introduction

Job Scheduler eases the management of ASE by providing the ability to define and schedule database administration and maintenance tasks. With Job Scheduler, jobs that normally require interaction from a database administrator can be scheduled to run unattended at the appropriate times, freeing the database administrator to attend to other issues.

Job Scheduler allows you to create and schedule jobs, and to share jobs and schedules. One database administrator can create a job, and other database administrators can then schedule and run that job on another server. You can create jobs as follows:

- From scratch using the command line or GUI

- From a SQL batch file

- From a template

Job Scheduler captures the results and output of jobs and records that information in log tables. This data can be viewed at a later date. In addition, Job Scheduler keeps a history of scheduled jobs; however, to keep a limit on the size of the history table, Job Scheduler monitors it and removes outdated, unnecessary history records.

Note To use Job Scheduler effectively, you should understand database management and administration. Job Scheduler executes operations that are usually performed by a database administrator.

Terminology and concepts

This section explains some of the basic concepts and terminology associated with using Job Scheduler.

Terminology

- A *job* is a series of one or more actions performed on a database in a single operation, such as backing up, updating statistics, and dumping a database.

- A *schedule* defines a timescale for how and when jobs are executed and re-executed.

- A *scheduled job* is a job that has been bound to a schedule. Only scheduled jobs are executed.

- The *Job Scheduler Task (JS Task)* is the functional component responsible for managing the schedules and providing timely notification to the Job Scheduler Agent to execute a particular job.

- The *Job Scheduler Agent (JS Agent)* is the functional component responsible for executing a job when notified by the JS Task.

- A *repeating schedule* is a schedule that is active more than once. All repeating schedules must have both start and end times.

- A *target server* is the Adaptive Server on which a job is scheduled to run.

- A *template* is a set of Transact-SQL (T-SQL) statements with parameters that can be used to create a job within the Job Scheduler.

Concepts

JS Task wakes up on demand, either by a job scheduled to be run or by a wake-up event that has occurred, such as the creation of a new job. At regular intervals (configurable by the database administrator), JS Task scans the *sybmgmtdb* database tables and collects applicable schedule information to maintain its dedicated server tables js_callouts and js_history. The JS Task performs the functions that allows Job Scheduler to perform its internal tasks. It is also what makes Job Scheduler run time programmable within ASE.

Job Scheduler components and functionality

Job Scheduler consists of the following components:

- An internal ASE task called the JS Task

- An external process called the JS Agent

- The sybmgmtdb database and stored procedures

- The graphical user interface (GUI) using the ASE plug-in in Sybase Central

- Predefined templates from which the database administrator can create and schedule useful, time-saving jobs

Basic tasks

The following list of tasks makes up the basic functionality of Job Scheduler and can be performed at the command line or in the Sybase Central graphical management tool:

- Create a job, schedule, or scheduled job

- Modify a job, schedule, or scheduled job

- Delete a job, schedule, or scheduled job

- Review job history

Job Scheduler architecture

The JS Task determines when scheduled jobs should run and creates a historical record of jobs that are run. It starts the JS Agent process and feeds it the necessary information to retrieve job information and run the job on the specified ASE.

When the JS Agent retrieves the job information from Job Scheduler's own database, called sybmgmtdb, it logs into the target ASE and issues the job commands. When the job completes, JS Agent logs any result or output to the log tables in the sybmgmtdb database.

All of the information about jobs, schedules, scheduled jobs, and data needed by the JS Task for internal processing is stored in the sybmgmtdb database, where data is usually accessed through stored procedures. The stored procedures make the data available to the GUI, the JS Agent, and the command-line interface. Only the JS Task accesses data directly from the sybmgmtdb database.

Using the data it receives, the GUI helps you create and schedule jobs, view job status and job history, and control jobs. It also provides an administration feature that turns the ASE internal task on and off and, therefore, allows Job Scheduler to process and execute scheduled jobs.

Templates are an important tool in defining parameterized tasks for self-management of the database, such as database backups, reorganization rebuilds, modification of configuration parameters, and statistics updates and monitoring. They are implemented as batch T-SQL commands for which you provide parameter values. Database administrators can use templates to generate jobs schedule them to run at specific times.

Figure 1-1 shows the various components that form the Job Scheduler architecture.

Figure 1-1: Job Scheduler architecture
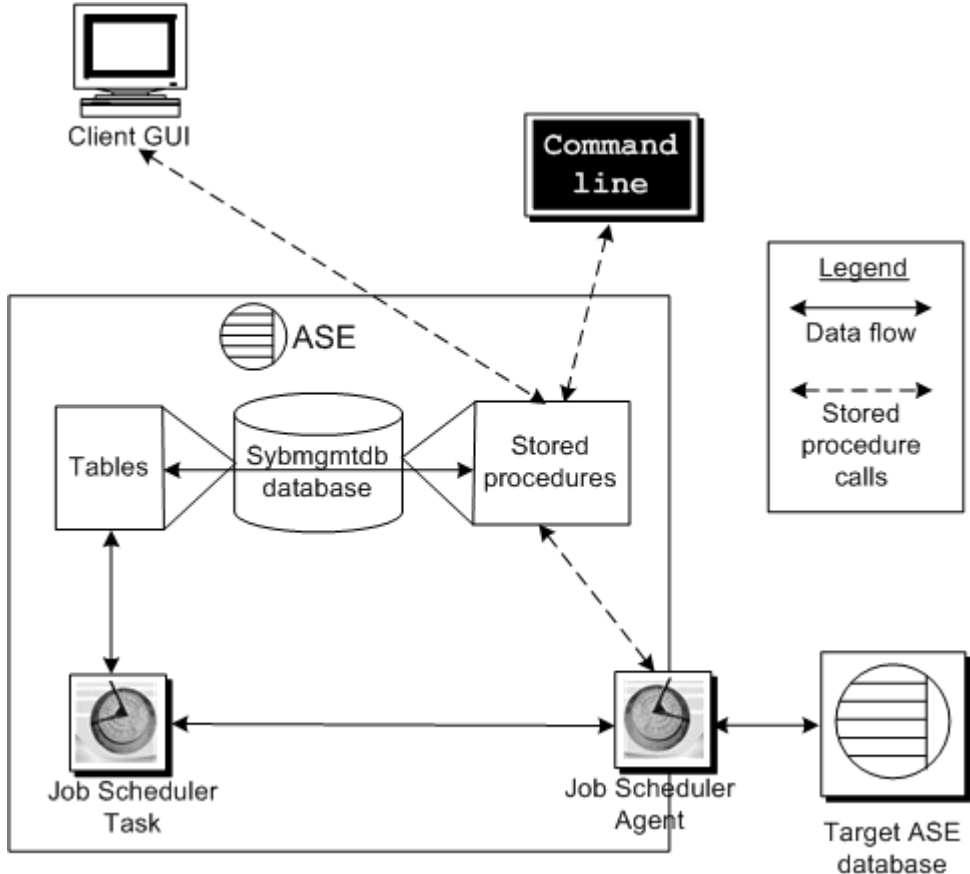
Process flow

As shown, the client GUI and the command line interface interact with the Job Scheduler database tables using stored procedures. Then, the stored procedures perform the actions requested by the user and maintain the definitions of the job commands, schedules, and other job information in the database tables.

The JS Task reads the job and schedule information from these tables and determines when a scheduled job needs to be executed. At the appropriate times, the JS Task informs the JS Agent of the job that needs to be run. Then the JS Agent retrieves the job information and executes it on the target ASE database, which can be either local (on the same server as Job Scheduler), or remote (anywhere outside the JS Agent, *including* on the same server Job Scheduler is installed on). The JS Agent collects the output and any result sets and stores this output in the appropriate tables in the Job Scheduler sybmgmtdb database.

5

Security

Job Scheduler uses predefined roles to manage security: js_user_role and js_admin_role.

js_user_role

A user with js_user_role has permission to create, modify, delete, and run jobs, schedules, and scheduled jobs using the Job Scheuler GUI or the command-line procedures. The owner of a job, schedule, or scheduled job, can limit which other users can see and use the object, but has no control over objects owned by another user.

js_admin_role

A user with js_admin_role has all the permissions of the js_user_role. It also allows a user to see and manipulate the jobs, schedules, and scheduled jobs of other users. Additionally, the js_admin_role allows a user to modify the following:

- the owner of a job, schedule, or scheduled job

- the amount of device space Job Scheduler consumes

- the frequency Job Scheduler checks for jobs

- the maximum log size a job can write

For more information on roles in ASE, see Chapter 1 of the *ASE System Administration Guide*, Volume 1.

Shared objects

By default, running a job is restricted to the *owner* of the entry in the scheduled jobs table. However, you can set the shared_run property for a scheduled job to make it available for other Job Scheduler users to run. Also, you can set it to make jobs and schedules to other users. This allows them to create their own scheduled jobs using these shared jobs and schedules. Sharing jobs, schedules, and scheduled jobs does not give other Job Scheduler users permission to modify or delete them; the sharing is read only.

Configuring and Running Job Scheduler

This chapter provides the necessary information to install, configure, and run Job Scheduler. It includes the following topics:

Topic	Page
Installing Job Scheduler manually	7
Setting up Job Scheduler users	9
Setting up access to target servers	10

Installing Job Scheduler manually

When you select the Job Scheduler option during the Adaptive Server installation, it is installed and configured. If you do *not* select Job Scheduler during the Adaptive Server installation, it will not be installed.

If you want to install Job Scheduler manually, after you have installed Adaptive Server, complete the following steps.

❖ **Installing Job Scheduler**

1 Create a device called sybmgmtdev with a size of at least 75 MB.

2 Run the *installjsdb* script:

```
isql –Usa –Psa_password –Sservername –i
$SYBASE/$SYBASE_ASE/scripts/installjsdb
```

Note You must have the directory with the location of the isql executable (*$SYBASE/$SYBASE_OCS/bin*) in your path.

The *installjsdb* script looks for the sybmgmtdb database. If it exists, the script creates Job Scheduler tables and stored procedures. If it does not exist, the script looks for the sybmgmtdev device on which to create the sybmgmtdb database, the tables, and stored procedures.

> **Note** If the *installjsdb* script finds neither a sybmgmtdev device nor a sybmgmtdb database, it creates a sybmgmtdb database on the master device. Sybase strongly recommends that you remove the sybmgmtdb database from the master device to prevent cluttering it and to make recovery easier in the case of a disk failure.

3   Create a directory services entry for the JS Agent using dscp, dsedit, or a text editor as appropriate for your specific directory services setup and operating system. Sybase suggests the name of "*server name*_JSAGENT":

```
servername_JSAGENT
    master tcp ether server_machine port_number
    query  tcp ether server_machine port_number
```

where:

*   *server_machine* is the name of the machine on which the Job Scheduler Adaptive Server is installed.

*   *port_number* is the port on which the JS Task communicates with the JS Agent.

    > **Note** You must specify a port currently not in use.

See the *System Administration Guide, Volume 1* for more information on directory services.

4   Use sp_addserver to create an entry in the sysservers table:

```
sp_addserver SYB_JSAGENT, null,
<servername_JSAGENT>
```

See the *ASE Reference Manual, Volume 2: Commands* for more information on sp_addserver.

5   Enable Job Scheduler:

```
sp_configure "enable job scheduler", 1
```

6   To start Job Scheduler, you can either restart the server or use the following commands:

```
use sybmgmtdb
go
sp_js_wakeup "start_js",1
go
```

Setting up Job Scheduler users

Set up users so they can create, manage, or execute jobs and schedules in the sybmgmtdb database.

Note Before you set up users in Job Scheduler, each user must have an ASE login.

❖ **Setting up users for Job Scheduler**

1 Add users to the sybmgmtdb database:

```
use sybmgmtdb
go
sp_adduser local_login_name
go
```

where *local_login_name* is the user's login name on the Job Scheduler server.

2 Grant the appropriate roles to users:

For Job Scheduler administrators:

```
sp_role 'grant', js_admin_role, local_login_name
go
sp_modifylogin local_login_name, 'add default role',
js_admin_role
go
```

For Job Scheduler users:

```
sp_role 'grant', js_user_role, local_login_name
go
sp_modifylogin local_login_name, 'add default role',
js_user_role
go
```

Setting up access to target servers

Because the Job Scheduler agent runs outside of Adaptive Server, target servers consider it a remote user. Consequently, you must define each target server (even when the target server is the server where the Job Scheduler is installed) and login to the *local* server, which is where Job Scheduler is installed.

❖ **Setting up access to target servers**

1   Create a directory services entry for each target server using dscp, dsedit or a text editor as appropriate to your directory services setup:

    ```
    target_servername
       master tcp ether targetserver_machine port_number
       query tcp  ether targetserver_machine port_number
    ```

    where:

    *   *target_servername* is the name of the server on which you want to schedule and run a job.

    *   *targetserver_machine* is the name of the machine on which you want to schedule and run a job.

    *   *port_number* is the port on which JS Agent connects to the target server.

    See the *System Administration Guide, Volume 1*, for more information on directory services.

2   On the Adaptive Server running Job Scheduler, create an entry in the sysservers table for each target server on which you want to run a job. A target server can be remote or it can be the Job Scheduler server itself. In both cases, you must add an remote entry for the target server in the sysservers table.

    *   **Remote target server:**   Use sp_addserver to add a remote target server. The server class you add must be ASEnterprise:

        ```
        sp_addserver target_servername, ASEnterprise,
        directoryservices_name
        go
        ```

        where:

        *   *target_servername* is an alias for the local server. When you install Job Scheduler with Adaptive Server, "loopback" is used as the alias for the server running Job Scheduler.

- *directoryservices_name* is the name used for the local server in the directory services file.

> **Note** A remote target server can be the same server as your Job Scheduler installation.

- **Local (the server running Job Scheduler) target server:**   The sysservers table already contains a local reference to the server where Job Scheduler is installed; however, to allow this server to be a target server, you must create a remote entry for the server in the sysservers table:

```
sp_addserver target_servername, ASEnterprise,
    directoryservices_name
go
```

where:

- *target_servername* is the server on which you want to schedule and run a job, in this case, the local (Job Scheduler) server.

- *directoryservices_name* is the name used for this target server in the directory services file.

3   Add logins for Job Scheduler users using sp_addexternlogin:

```
sp_addexternlogin target_servername, localname,
remotename, remotepwd
go
```

where:

- *target_servername* is the server on which you want to schedule and run a job.

- *localname* is the user's login name on the Job Scheduler server.

- *remotename* is user's login name on the target server.

- *remotepwd* is user's password on the target server.

> **Note** You must add external logins for all users for all target servers, even if the target server is the server where Job Scheduler is installed.

You can now create jobs, schedules, and scheduled jobs using stored procedures from the command line or using the Sybase Central ASE plug-in. See Chapter 5, "Command Reference," for more information on Job Scheduler stored procedures. See Chapter 6, "Managing Job Scheduler in the Sybase Central ASE Plug-in," for more information on using Job Scheduler with Sybase Central.

CHAPTER 3   **Using Templates to Create Jobs**

This chapter describes the installation and use of templates with Job Scheduler.

Introduction

Templates provide one more way to ease the management of your Adaptive Server databases. A template helps database administrators to create T-SQL scripts that perform specific database tuning and maintenance tasks; the T-SQL that it creates becomes the T-SQL text for the job itself. These jobs can subsequently be customized to a particular server and scheduled to run on any network server at any specified time. As a result, templates can save database administrators the work of creating their own T-SQL or cron scripts, freeing them to concentrate on other duties.

When you use the Sybase Central ASE plug-in, a wizard leads you through each step to create the T-SQL that invokes the procedures for managing your databases.

Note When you create a job from a template that uses stored procedures, the stored procedures must exist on the server on which the job runs. See "Installing stored procedures on a target server" on page 14 for more information.

Installing stored procedures on a target server

Before you can run jobs on target servers, you must install the stored procedures used by the templates on the target servers where you want your jobs to run.

❖ **Installing stored procedures on a target server**

1 In your Sybase installation, go to the directory that contains the installation utility, which is *$SYBASE/$SYBASE_ASE/jobscheduler/Templates/sprocs*.

2 Run installTemplateProcs with the following parameters:

   • On UNIX:

   ```
   installTemplateProcs target_servername username
   password
   ```

   • On Windows:

   ```
   installTemplateProcs.bat target_servername
   username password
   ```

   where:

   • *target_servername* is the name of the target server.

   • *username* is the login name on the target server for the user installing the templates.

   • *password* is the password on the target server for the user installing the templates.

The stored procedures are now installed in the sybsystemprocs database on the target server. See Chapter 5, "Command Reference" for more information on stored procedures.

Installing the templates into Job Scheduler

Before you can use the templates, you must install the templates into the Job Scheduler you select to manage your jobs.

❖ **Installing the Job Scheduler templates**

1 In your Sybase installation, go to the directory containing the installation utility, which is *$SYBASE/$SYBASE_ASE/jobscheduler/Templates/xml*.

2   Run installTemplateXML with the following parameters:

- On UNIX:

      ```
      installTemplateXML servername machinename
      serverport username password language
      ```

- On Windows:

      ```
      installTemplateXML.bat servername machinename
      serverport username password language
      ```

where:

- *servername* is the name of the server where Job Scheduler is installed.

- *machinename* is the name of the machine hosting the Job Scheduler server.

- *serverport* is the port number for connecting to the Job Scheduler server.

- *username* is the user's login name on the Job Scheduler server.

- *password* is the user's password on the Job Scheduler server.

- *language* is the code to indicate which language version of the templates is installed.

   Job Scheduler supports the following languages. Use the following language codes to specify language:

   - English — en

   - French — fr

   - Japanese — ja

   - Korean — ko

   - Simplified Chinese — zh

   If you do not specify a language code, Job Scheduler defaults to English templates.

The Job Scheduler templates are now installed on the Job Scheduler server.

Using Job Scheduler templates

The following templates are available to help you create jobs.

Backup

By allowing you to easily create database backup jobs, the backup templates help you preserve your data.

• **Sybase backup database to disk template**   This template uses the dump database command to allow you to back up one or more databases to disk. When creating a job from this template, you specify the database name and dump directory. You can also specify if you want to compress or stripe the database, and if you want to include the server name and date in the dump file name.

• **Sybase backup transaction log to disk template**   This template uses the dump transaction command to allow you to back up the transaction log for one or more databases. When creating a job from this template, you specify the database name and dump directory. You can also specify if you want to compress or stripe the database, and if you want to include the server name and date in the dump file name. Additionally, you can specify a time threshold so if that amount of time has passed since the last dump, the log will be dumped. Similarly, you can specify a row threshold to force a dump of the log if it contains at least that number of rows. If you do not provide values to any of these parameters, the log dump occurs.

Statistics management

The statistics management templates help you keep statistics current, so you can generate efficient query plans.

• **Sybase server update statistics template**   This template uses the update statistics command to allow users to update statistics on all tables in all server databases. The template allows you to supply values for the various options of the update statistics command, while the table, column, and index values are retrieved from the system tables.

Additionally, you can provide threshold values for data change, page count, and row count. If you supply any of these optional values, they will be used as a gauge in determining whether the update statistics command should be run. If current values for any one of these thresholds meet or exceed the threshold for a given table, update statistics will be executed on that table. After update statistics is run, the sp_recompile procedure is run for the table.

- **Sybase update statistics template**   This template uses the update statistics command to allow you to update statistics at the table, partition, column, and index levels. Additionally, you can provide threshold values for data change, page count, and row count. Data change is a metric that allows you to track the amount of changes to a table, column, or partition. You can use it as an indicator that the update statistics command might help improve performance.

  If you provide threshold values, they determine if the update statistics command should be run. After update statistics is run, the sp_recompile procedure is run for the table you specified in the template wizard.

- **Sybase delete statistics template**   This template uses the delete statistics command to allow you to delete statistics for all columns in the specified table, for a specific column in the table, or for a partition and local indices on a partition. You must supply the database and table name. You can also specify a column list of which statistics you want to delete. If the column list is omitted, statistics for columns in the entire table are deleted.

Reorganization

The reorganization templates help you keep your database organized, avoiding lost table space and noncontiguous table data.

- **Sybase rebuild indexes template**   This template runs the reorg rebuild command on an index or indexes for a table. You must specify the database, table, and indexes required by the reorg rebuild command. You can also specify an index partition name and a database backup after the reorg rebuild command executes.

- **Sybase rebuild tables template**   This template runs the reorg rebuild command on the entire table. You must specify the database and tables required by the reorg rebuild command. You can also specify to back up the database after the reorg rebuild command executes.

- **Sybase reclaim index space template**   This template reclaims unused space left on a page as a result of deletions and updates that shorten rows. It invokes the reorg reclaim space command to reorganize an *index's* data pages. If you specify more than one index, space is reclaimed for each index. You can also specify the name of the index partition that contains the specified index.

- **Sybase reclaim table space template**   This template reclaims unused space left on a page as a result of deletions and updates that shorten rows. It invokes the reorg reclaim space command to reorganize a *table's* data pages. If you specify more than one table, space is reclaimed for each table. You can provide a partition name to limit the reorg reclaim space command to that partition of the table on the specified partition. You can also specify a partition name to limit reorganization to a partitioned portion of a table.

Reconfiguration

The reconfiguration templates help you keep user connections, metadata cache, and locks configured properly to meet your current demands.

- **Sybase reconfigure locks template**   This template allows you to set up automatic reconfiguration for the number of locks allowed by the server at any one time, based on server data and user input. You can specify the number of locks you want to add.

- **Sybase reconfigure metadata cache template**   The metadata cache is a reserved area of memory that is used for databases, indexes, and objects. Using this template, you can create jobs to determine the proper number of each metadata object type for your server. While creating the job, you supply values that are used at runtime to determine the proper number of objects needed for your current server conditions. The job will determine the current number of active objects and adjust the allowable number of objects of the specified type.

  Alternatively, you can specify the exact number for one or all metadata object types, bypassing the self-tuning feature of this template.

- **Sybase reconfigure user connections template**   This template calls sp_configure to set the number of user connections. You can provide the new number of user connections, or the template can calculate it using information you supplied in the wizard.

Template updates

You can download updates to the Job Scheduler templates and new templates at Sybase CodeXchange on the Internet.

❖ **Getting updates and new templates**

1   Log in to MySybase at www.sybase.com.

2   Log in to CodeXchange.

3   Click the Projects tab.

4   Click the ASE project in the projects box.

5   Scroll to the bottom to the subprojects. Select Job Scheduler.

Note Each release of Adaptive Server includes all the available Job Scheduler templates.

Adaptive Server Enterprise

CHAPTER 4 **Using Job Scheduler at the Command Line**

This chapter provides examples for creating and scheduling jobs from the command line. You can also perform these procedures from the ASE plug-in in the Sybase Central management tool. See Chapter 6, "Managing Job Scheduler in the Sybase Central ASE Plug-in," for more information.

Topic	Page
Creating a job	21
Creating a schedule	24
Creating a scheduled job	25
Deleting a scheduled job	26
Modifying a scheduled job	26
Invoking stored procedures on a target server	27
Managing jobs	27
Managing job history and logs	28

Creating a job

Following are the steps to create a job that prints Hello World.

❖ **Creating a Hello World job and a list of its history**

1 Perform the server and user setup described in Chapter 2, "Configuring and Running Job Scheduler." Be sure you have permission to log in to your target server using the user name and password that was supplied to the sp_addexternlogin stored procedure.

2 In the following code example, use the sp_sjobcreate procedure to create a job:

```
use sybmgmtdb
go
declare @jobcmd varchar(255), @jobid int
select  @jobcmd='jcmd=print "Hello World.",server=
YOUR_SERVER'
+ ',starttime=' + convert(varchar(32),getdate())
exec @jobid=sp_sjobcreate 'sjname=hello', @jobcmd
go
```

where

YOUR_SERVER is the name of your target server

3 Use sp_sjobhelp to look at the scheduled job that you created:

```
> exec sp_sjobhelp 'sjname=hello'
> go
```

4 The following summary of the scheduled job is returned:

```
sjob_id: 127  name: 'hello'
owner           : jsadmin1
created         : Jul 14 2005 4:42AM
state           : enabled
job name        : 114 - 'job_114'
schedule name   : 115 - 'sched_115'
server          : pgibson_js
-- job --------:
description     :
owner           : jsadmin1
created         : Jul 14 2005 4:42AM
-- schedule ---:
description     :
owner           : jsadmin1
created         : Jul 14 2005 4:42AM
starttime       : 04:42
startdate       : 14 Jul 2005
```

5 Use sp_sjobhistory to look at the short list history of executing a scheduled job:

```
sp_sjobhistory 'sjname=hello', @option='list_short'
go
```

6 The following short list is returned:

```
sjob_id sjob_jobname  sjob_schedname  sjob_server   sjob_state
------- ------------  --------------  -----------   ----------
127     job_114       sched_115       pgibson_js    C2
```

```
sjob_start                 sjob_user_run   sjob_user_req  sjob_size
----------                 -------------   -------------  ---------
Jul 14 2005 4:42AM   jsadmin1            jsadmin1        51
```

7   Use sp_sjobhistory to look at the output from executing a scheduled job:

```
sp_sjobhistory 'sjname=hello', 'list_output'
go
```

8   The following output is returned:

```
 jsout_run_id   jsout_seqno   jsout_size     jsout_text
------------   -----------   -----------   ----------
144           0            51             Changed database context to 'master'.

hello world
```

Job state codes

The following table defines the job state codes that are returned in the history table.

Table 4-1: Job state codes descriptions

State name	State code	Description
waiting	W	An initial state for a job that has been created or is waiting to schedule.
queued	Q	Job Scheduler can run the job, but it may be waiting for a free thread to run it.
busy	B	Job Scheduler did not start the job because this job was already running.
runable	R1	Job Scheduler has started the job.
running	R2	The job is running.
completing	C1	The job has completed, and Job Scheduler is cleaning up logs, history, and so on.
completed	C2	The job has completed. Post-processing SQL is complete.
terminating	T1	Job Scheduler is terminating the job, killing the thread.
terminated	T2	Job Scheduler has terminated the job. Post-processing SQL is complete.
timing-out	X1	The job has timed out, and Job Scheduler is terminating the job.
timed-out	X2	The job has timed out and been terminated.

State name	State code	Description
missed	M	The scheduled run of the job was missed when Job Scheduler was inactive.

Creating a schedule

The following code example shows how to create a schedule in Job Scheduler on the command line. You can perform the same task using the ASE plug-in in the Sybase Central management tool.

When you create a repeating schedule, it must have a start time *and* an end time.

Note When you create a name for a schedule, the name must begin with a letter. If you create a name beginning with a digit, an error occurs.

The following example creates a schedule that operates every five minutes, between 08:00 and 18:00, and is valid every day of the week:

```
sp_sjobcreate @name='sname=every5m_8to6',@option=
'repeats=5minutes, starttime=08:00am, endtime=18:00'
```

The following example creates a schedule that operates every hour on Saturdays and Sundays between 08:00 and 18:00:

```
sp_sjobcreate @name='sname=hourly_8to6_weekends',
@option='repeats=1hour,starttime=08:00am,endtime=
18:00,days=Saturday:Sunday
```

The following example creates a schedule that operates at 04:00am on the first and last day of every month, where 32 is the last day of the month:

```
sp_sjobcreate @name='sname=run_4am_1st_and_last',
@option='starttime=04:00,endtime=04:00,dates=1:32'
```

The following example creates a schedule that operates at 09:00, beginning the first of January and ending the first of February:

```
sp_sjobcreate @name='sname=run_daily_9am_Jan',
@option='starttime=09:00,endtime=09:00,repeats=
1day,startdate=1 January 2005,enddate=1 February 2005'
```

Note The schedule start time is inclusive—a job will begin at the start time. The schedule end time is exclusive—a job will run up to but not include the end time. For example, if a schedule has a start time of 13:00 and an end time of 16:00 and repeats every hour, it run at 13:00, 14:00, and 15:00, but not at 16:00.

Creating a scheduled job

The following code example shows how to create a scheduled job in Job Scheduler on the command line. You can perform the same task using the ASE plug-in in the Sybase Central management tool.

Note When you create a name for a scheduled job, the name must begin with a letter. If you create a name beginning with a digit, an error occurs.

❖ **Creating a scheduled job that gathers statistics on a trial basis for one month**

1   Create the schedule:

```
sp_sjobcreate @name='sname=Aug_stats_trial',
@option='starttime=23:00,endtime=23:00,repeats=
1day,startdate=1 August
2005,enddate=31 August 2005'
```

2   Create the job:

```
sp_sjobcreate @name='jname=new_stats_proc',
@option='jcmd=''exec statsdb..
new_stats_proc'',jdesc=New statistics
proc.,jproperties=shared'
```

3   Create a scheduled job on a server called "devtest1":

```
sp_sjobcreate @name='sjname=new_stats_devtest1',
@option='sname=Aug_stats_trial,jname=
```

```
new_stats_proc,server=devtest1'
```

> **Note** The server name assigned to the scheduled job is the network name (the *pname* property in sp_addserver) in the interfaces file, not the logical name defined in the sysservers table.

Deleting a scheduled job

Here is an example that deletes a scheduled job.

❖ **Deleting a scheduled job**

• Use sp_sjobdrop to delete a scheduled job called "new_stats_devtest1":

```
sp_sjobdrop 'sjname=new_stats_devtest1'
go
```

Modifying a scheduled job

Using sp_sjobmodify you can configure a scheduled job to run at repeating intervals, on days of the week, or on specific dates in a month. The following examples modify the "hello world" job.

The following code sets the job to run every day at 09:00:

```
sp_sjobmodify 'sjname=hello', 'starttime=
09:00,repeats=1day'
go
```

The following code sets the job timout property to 120 minutes::

```
sp_sjobmodify @name='jname=hello',
@option='default_timeout=120'
go
```

The following code sets the job to run at 09:00 on Mondays and Fridays:

```
sp_sjobmodify 'sjname=hello', 'starttime=
09:00,endtime=09:00,days=Monday:Friday'
go
```

The following code changes the target server for the scheduled job:

```
sp_sjobmodify 'sjname=hello', 'server=prodASE
go
```

Invoking stored procedures on a target server

Scheduled jobs often will invoke stored procedures on the target server to perform the required function. The following example shows a scheduled job invoking sp_who:

```
use sybmgmtdb
go
declare @jobcmd varchar(255), @jobid int
select  @jobcmd='jcmd=exec sp_who,server=YOUR_SERVER'
+ ',starttime=' + convert(varchar(32),getdate())
exec @jobid=sp_sjobcreate 'sjname=hello', @jobcmd
go
```

where *YOUR_SERVER* is the name of your target server.

Note When Job Scheduler executes a job it prefixes the job text with SQL that creates the job runid and the scheduled job ID. Therefore, the call to sp_who needs to use exec to invoke the stored procedure.

Managing jobs

Using sp_sjobcontrol, you can perform the following ad hoc actions on Job Scheduler:

- Terminate a running job

- Run a scheduled job immediately

- Enable and disable a scheduled job

See sp_sjobcontrol on page 41 for more information.

Managing job history and logs

You can adjust the configuration, such as, the rules and thresholds, that controls the automatic management of the history and job output logs.

To adjust these configuration parameters using the ASE plug-in in Sybase Central, see "Administering Job Scheduler" on page 50.

CHAPTER 5   **Command Reference**

This chapter describes the properties and parameters of the command line stored procedures.

Command line stored procedures

Adding, modifying and deleting

There are two methods for creating a scheduled job: by specifying both the job and schedule information in one operation, or by defining the job and the schedule separately and then combining them into a scheduled job.

Reporting and listing

Stored procedures are provided for the command line interface to obtain information about the jobs and schedules configured in the Job Scheduler database. The information can be provided in a formatted report or as a simple list.

Running jobs

You can run jobs by scheduling them or using ad hoc commands.

Jobs history and output

It is possible to manage job history and job output separately. Deleting job history also deletes any corresponding job output, but deleting job output records does not delete job history. This flexibility allows for different rules to govern how much job history and job output is retained.

Job Scheduler users without the js_admin_role are restricted to viewing and manipulating the job history and job output from jobs which have their user name recorded in the history entry. However, a user with js_admin_role can use the all option to override this limitation and extend the scope to the history and output to all Job Scheduler users.

Table 5-1: Job Scheduler stored procedure descriptions

Procedure name	Procedure purpose
Create, modify, delete	
sp_sjobcreate	Creating scheduled jobs, jobs or schedules
sp_sjobcmd	Managing the SQL command text of a job
sp_sjobmodify	Modifying scheduled jobs, jobs and schedules
sp_sjobdrop	Removing scheduled jobs, jobs and schedules
Reporting, listing	
sp_sjobhelp	Reporting and listing of scheduled and running jobs
Running jobs	
sp_sjobcontrol	Job Scheduler administration and control of scheduled or running jobs
Jobs history, output	
sp_sjobhistory	Viewing and administration of the job history and output

Command syntax

Job Scheduler provides a concise set of stored procedures for working with scheduled jobs. Their names are prefixed with sp_*sjob*, where *sjob* is an abbreviation for scheduled job.

The stored procedures for creating jobs, schedules, and scheduled jobs will use the server user name to record the owner. The suser_name system function will be used to obtain this.The stored procedures also support working with the underlying job and schedule objects, and provide an interface to control scheduled jobs that are currently running and manage the history and job output that they produce.

The stored procedures accept a name or ID argument, which specifies the object they should operate on. Stored procedures that operate on several different objects, such as a job, a schedule, or a scheduled job, accept a prefix to the name or ID argument, which specifies the sort of object the name or ID refers to: for schedules – @sname; for jobs – @jname.

The following is an example of @sname:

```
@name= 'sname=daily_schedule'
```

The following is an example of using @jname:

```
@name= 'jname=run_update_stats'
```

The name or ID default is to reference a scheduled job.

General usage

- Command names, command keywords, the word "null", data items, and statistic types are not case sensitive. File names, view names, and other user-supplied names are case sensitive.

- If a parameter value contains embedded spaces (such as those in data items, statistic types, and date-time specifications), you must surround the value with quotes. Matched pairs of single-quote or double-quotes are valid delimiters.

- If the parameter value contains an embedded quote that is the same as the character used to delimit the entire value, supply a pair of the quotes within the parameter value. Job Scheduler compresses the pair of quotes to a single character.

- The word "null" within quotes is not considered a keyword.

- You can enter Job Scheduler commands on multiple lines.

sp_sjobcreate

Description

This procedure serves several purposes:

- When only job information is supplied, it creates a new job.

- When only schedule information is supplied, it creates a new schedule.

- When job and schedule information is supplied, it either combines an existing job and schedule to create a new scheduled job, or creates a new job and a new schedule, and combines them into a new scheduled job.

By default, the @name argument is the name of a scheduled job. To specify the name of a job or a schedule, the @name argument is prefixed with jname or sname.

Syntax
sp_sjobcreate @name='*jsname*', @options='server, jname, jdesc, jcmd, sname, sdesc, repeats, properties, starttime, enddate, endtime, days, dates'

Parameters
name (jname, sname, sjname)
The name of the new job, schedule, or scheduled job.

Note When you create a name for a job, a schedule, or a scheduled job, the name must begin with a letter. If you create a name beginning with a digit, an error occurs.

---

option
A list separated by commas of the field names and values you use to create a job, a schedule, or a scheduled job. Values are:

- server – the name of the server where the job runs. The default is the local server.

- jname – the name of the job, which must be unique.

- jdesc – comments describing the job.

- jcmd – the SQL text, used for simple jobs when it is easy to provide the text directly.

- sname – the name of the schedule, which must be unique.

- default_timeout – the maximum amount of time permitted for the execution of a job. This value is used by scheduled jobs if the scheduled job's timeout property is not set.

- sdesc – comments describing the schedule.

- timeout – the maximum amount of time a scheduled job can execute. This value supercedes the job's default_timeout value

- repeats – the interval at which the schedule repeats. 0 or NULL means it does not repeat. The value is a number followed by:

  - day or d

  - days or dd

  - hour or h

  - hours or hh

- minute or m

- minutes or mm

- properties – the properties of the job, the schedule, or scheduled job:
  jproperties, sproperties, or sjproperties, respectively.

- startdate – the date on which a schedule becomes active.

- starttime – the time of day when a scheduled job begins.

- enddate – the date on which a schedule becomes inactive.

- endtime – the time at which a schedule becomes inactive for the
  remainder of the day.

- days – a list of days separated by colons. Full names or abbreviations in
  the locale of the server may be used.

- dates – a list of dates in a month, from 1 to 31, separated by colons. The
  number 32 represents the last day of any month.

Note An error occurs if you try to specify values for both days and
dates.

Return value
- The job ID for a new job.

- The schedule ID for a new schedule.

- The scheduled job ID for a new scheduled job.

- An error code.

Examples
Example 1 The following example creates a new job called "find_old_logins"
that consists of running a stored procedure:

```
sp_sjobcreate @name='jname=find_old_logins'
@option='jcmd="exec
sp_find_old_logins",jproperties=shared'
```

Example 2 The following example creates a new schedule that becomes valid
at 1:00 am and repeats every day:

```
sp_sjobcreate @name='sname=daily 01:00am',
@option='repeats=1day,starttime=01:00am,
endtime=02:00am'
```

Example 3 The following example creates a new scheduled job to run on
server "dev1" using the existing job "find_old_logins" and schedule called
"daily 01:00am":

```
sp_sjobcreate @name='dev1_old_logins',
@option='server=dev1,jname=find_old_logins,
sname=daily 01:00am'
```

Example 4 The following example creates a new schedule called "evening_sales_report," using new job called "load_sales_data," and a new schedule that runs every Monday, Wednesday and Friday at 23:00. The schedule is given a default name based upon the ID value in the schedule table.

```
sp_sjobcreate @name='evening_sales_report'
@option='server=reports, jname=load_sales_data,
jcmd=exec sp_new_sales_data,
starttime=23:00,endtime=23:00,days=Monday:Wednesday:Fr
iday'
```

Usage

- A schedule using both a days value and a dates value is not allowed. However, separate schedules using days and dates can be bound to the same job.

- A schedule using a repeats value of more than one day is not valid with days or dates values.

- A schedule using a repeats value equal to one day is not valid with days or dates values. By default all days are valid when repeats equals one day.

- A schedule using a repeats value of less than one day and a days value of NULL is, by default, valid on all days.

- If the startdate is equal to the enddate and the endtime is specified, the endtime value must be equal to or later than the starttime value.

- For schedules with a repeats value of one day or more, the endtime value has no meaning and is ignored.

- If endtime is not specified, the default is midnight.

- If starttime is not specified, the default is 00:00am.

sp_sjobcmd

Description

Allows the SQL text of a job to be manipulated. When @option='list', the SQL text is listed.

By default, the @name argument is the name or ID of a scheduled job. To specify the name or ID of a job, the @name argument is prefixed with jname.

Syntax	sp_sjobcmd @name='…', @option='…', @text='...'
Parameters	*name*

> The name or ID of a scheduled job or job. If it is prefixed with jname, the name or ID of a job.

option

> An option to select add, list, or drop to the SQL text for a job.
>
> - drop – deletes all SQL text for the specified job.
>
> - list – displays the job command text stored in the database.
>
> - add – appends to any existing SQL text for the specified job, allowing large SQL batches to be stored in several chunks. When JS Agent runs the job, the SQL text for the job is concatenated.

text

> The SQL text to store.

Return value	Returns 0 or an error code.
Examples	**Example 1** The following example lists the SQL text for the job that is referenced by the scheduled job called "svr1_check_stats":

```
sp_sjobcmd 'sjname=svr1_check_stats', 'list'
```

Example 2 The following example deletes the existing SQL text for the "load_sales_data" job and then stores new SQL text for the job:

```
sp_sjobcmd 'jname=load_sales_data', 'drop'
sp_sjobcmd 'jname=load_sales_data', 'add',
@text='truncate table  sales_report_data'
go
'
```

Usage	Job Scheduler does not implicitly add a new line at the end of the job text. So, when you create a job with sp_sjobcmd, you must explicitly add a new line after you execute a call:

```
sp_sjobcmd 'jname=load_sales_data', 'drop'
sp_sjobcmd 'jname=load_sales_data', 'add',
@text='truncate table  sales_report_data'
go
'
```

sp_sjobmodify

Description   Allows you to modify any of the fields in a scheduled job, a job, or a schedule.

Syntax   sp_sjobmodify @name='...', @option='...'

Parameters   *name*
    The name or ID of the scheduled job, job, or schedule that is to be modified.

    option
    A comma-separated list of the field names to modify and the new values:

- sjname – a new name for the scheduled job that must be unique. A new ID may not be supplied.

- enable – 1 enables the scheduled job, 0 disables it.

- sjproperties – the properties of the scheduled job.

- sjowner – the owner of the scheduled job. Caller must have js_admin_role to change the owner.

- server – the server where the scheduled job should run.

- timeout – a timeout value specified for the scheduled job and server, in minutes.

- locale – the locale to be used by the client connection when the scheduled job runs.

- jname – a new name for the job; this must be unique. A new ID may not be supplied.

- jdesc – comments describing the job.

- jcmd – SQL text used for simple jobs when it is convenient to supply the text directly. Replaces all existing SQL text.

- jproperties – the properties of the job.

- jowner – the owner of the job. Caller must have js_admin_role to change the owner.

- default_timeout – a timeout value for the job, in minutes.

- sname – a new name for the schedule; this must be unique. A new ID may not be supplied.

- sdesc – comments about the schedule.

- sowner – must have js_admin_role to change the owner.

- sproperties – schedule properties.

Adaptive Server Enterprise

- • reset – "true" resets the timing information in the schedule. It clears the following: repeat, units, startdate, starttime, enddate, endtime, days, and dates.

- • repeats – the interval at which the schedule repeats. NULL or 0 means it does not repeat. The value is a number followed by:

  - • day or d

  - • days or dd

  - • hour or h

  - • hours or hh

  - • minute or m

  - • minutes or mm

- • startdate – the schedule becomes active on this date. The time part is ignored.

- • enddate – the schedule becomes inactive. The time part is ignored.

- • starttime – the time of day when an active schedule begins to operate. The date part is ignored. If starttime is not specified, it defaults to midnight (00:00).

- • endtime – the time of day when an active schedule ceases to operate. The date part is ignored. If endtime is not specified, it defaults to midnight (00:00).

- • days – a colon-separated list of days. Full or abbreviated names in the locale of the server may be used.

- • dates – a colon-separated list of dates in a month.

Return value   Returns 0 on success or an error code.

Examples   **Example 1** The following example modifies the scheduled job "svr1_clean_stats," updating the schedule to repeat every two hours between 10:00 and 16:00. The days or dates on which the scheduled job operates are unchanged.

```
sp_sjobmodify @name='svr1_clean_stats',
@option='repeats=2hours,starttime=10:00,endtime=16:00'
```

Example 2 The following example modifies the job "backup_db5," updating the default timeout to 120 minutes:

```
sp_sjobmodify @name='jname=backup_db5',
@option='default_timeout=120'
```

Example 3 The following example modifies the job named "orders_picked_report," removing the shared property and changing the owner to mary. The caller requires js_admin_role to change the job owner.

```
sp_sjobmodify @name='jname=orders_picked_report',
@option='jproperties=shared:false,owner=mary'
```

Usage

By default, the @name argument is the name or ID of a scheduled job. To specify the name or ID of a job or a schedule, the @name argument is prefixed with jname or sname.

Following is an example of using @jname:

```
@name= 'jname=run_update_stats'
```

Following is an example of using @sname:

```
@name= 'sname=daily_schedule'
```

sp_sjobdrop

Description

Deletes a job, a schedule, or a scheduled job.

By default, the name argument is the name or ID of a scheduled job. To specify the name or ID for a job or a schedule the name argument is prefixed with jname or sname.

Syntax

sp_sjobdrop @name='...', @option='...'

Parameters

name
 The name or ID of the scheduled job, job, or schedule to delete.

option

A list of options for the command:

- all – used with a job or schedule to specify that all scheduled jobs owned by the caller that are using the specified job or schedule should also be deleted. When used with a scheduled job, it specifies that the job and schedule associated with the scheduled job should also be deleted, provided that no other scheduled jobs reference them.

- all_users – allows a user with the js_admin_role to use sp_sjobdrop on scheduled jobs, jobs, and schedules that are owned by someone else.

- show – displays the jobs, schedules, and scheduled jobs that will be deleted by this call to sp_sjobdrop, before any deleting occurs.

- force – allows you to remove jobs from the database, even if they are currently running.

(See the Usage section for more details on these parameters.)

Return value
Returns 0 on success or an error code.

Examples
Example 1 The following example deletes the scheduled job called "svr1_clean_stats":

```
sp_sjobdrop 'svr1_clean_stats'
```

Example 2 The following example deletes the schedule called "daily 01:00am":

```
sp_sjobdrop 'sname=daily 01:00am'
```

Example 3 The following example deletes the job called "load_sales_data":

```
sp_sjobdrop 'jname=load_sales_data'
```

Example 4 The following example deletes the job called "load_sales_data" and any scheduled jobs the caller owns that were using it.

```
sp_sjobdrop @name='jname=load_sales_data',
@option='all'
```

Usage
- An error occurs if you attempt to delete a scheduled job that is currently running, unless you use the force option.

- By default, the sp_sjobdrop procedure only operates on the scheduled job, job, or schedule specified by the @name argument.

For the general administration of jobs and schedules, the all and all_users options are provided. The following describes these options:

- When a scheduled job is specified and the all or all_users option is not used, only the specified scheduled job is deleted.

- When the all option is used for a scheduled job, the scheduled job will be deleted. If there are no more references to the job that will also be deleted, and if there are no more references to the schedule, then the schedule will also be deleted.

- When a job or a schedule is specified and the all or all_users option is not used, an error occurs if the job or schedule is referenced by one or more scheduled jobs.

- When the all_users option is used for a specified job or a schedule and the scheduled jobs that reference the job or schedule are owned by the caller, then the job or schedule and the referencing scheduled jobs are deleted.

- When the all_users option is used for a specified job or a schedule and the caller has the js_admin_role, then the job or schedule and all referencing scheduled jobs are deleted.

The show option is provided so you can see which scheduled jobs, jobs, and schedules would be deleted by a call to sp_dropjob—without actually deleting them.

sp_sjobhelp

Description	Generates a list or report about all the scheduled jobs, jobs, or running jobs that are visible to the caller, or about a restricted set based on the name and the option values.
Syntax	sp_sjobhelp @name='...', @option='...'
Parameters	*name* The name or ID of the scheduled job, a job, or a runid. The @name argument restricts the scope to a single scheduled job, all scheduled jobs using a particular job, or a single run of a scheduled job. By default, @name is the name ID of a scheduled job. To specify the name, the ID of a job, or the runid, the @name argument is prefixed with jname or runid.

option

> Specifies a comma-separated list of options, the option_name=option_value pairs that define the action to perform, and filters to restrict the information returned.

- report – format as a report.

- list – format as a list.

- running – produce a report of running jobs.

- scheduled – produce a report of scheduled jobs.

- unscheduled – produce a report of unscheduled jobs.

- all_users – include information about all users.

- owner – restrict to information owned by this user.

- jobs – provide information about jobs.

- schedules – provide information about schedules.

- user – restrict information for scheduled jobs to run on behalf of this user. Applies only to running jobs.

Return value   Returns 0 on success or an error code.

Examples   **Example 1** The following example produces a list of all the scheduled jobs that are running:

```
sp_sjobhelp  @option='list,all_users,running'
```

Example 2 The following example produces a report showing all the jobs scheduled by the caller:

```
sp_sjobhelp @option='report,scheduled'
```

Example 3 The following example produces a list of all the jobs and schedules that are not used in any scheduled jobs:

```
sp_sjobhelp  @option='list,all_users,unscheduled'
```

Usage   The report will be in several sections, showing the scheduled jobs, jobs that do not have schedules, and the schedules.

sp_sjobcontrol

Description   Defines an interface for controlling running jobs.

Syntax	sp_sjobcontrol @name='...', @option='...'
Parameters	*name*

> The name or ID of the scheduled job, a job, or a runid to be controlled.

option
> Specifies a comma-separated list of options and option_name=option_value pairs that define the action to perform.

- terminate – instances of the job with the given name, ID, or runid are terminated.

- enable – instances of the scheduled job with the given name or ID are enabled.

- disable – instances of the scheduled job with the given name or ID are disabled. Any currently running jobs are unaffected.

- run_now – schedules the job with the given name or ID to run now.

- all_users – allows users with the role js_admin_role to administer scheduled jobs owned by other users.

- start_js – starts Job Scheduler.

- stop_js – stops Job Scheduler and terminates any running jobs.

- stop_js_wait – waits for running jobs to complete, and then stops Job Scheduler.

- stop_js_timeout=*minutes* – waits a specified number of minutes, and then stops Job Scheduler, terminating any running jobs.

Return value	Returns 0 for success or an error code.
Examples	**Example 1** The following example requests the JS Task and JS Agent to terminate the running job with runid 1532:

```
sp_sjobcontrol @name='runid=1532', @option='terminate'
```

Example 2 The following example marks the scheduled job called "svr1_load_sales_data" so that it can be run now. The existing schedule is unchanged.

```
sp_sjobcontrol @name='svr1_load_sales_data',
@option='run_now'
```

Example 3 The following example disables all of the caller's scheduled jobs that use the job called "check_user_logins".

```
sp_sjobcontrol @name='jname=check_user_logins',
@option='disable'
```

Usage   By default, @name is the name, the ID, or a scheduled job. To specify the name
                     or ID of a job or as a runid, the @name argument is prefixed with `jname` or
                     `runid`.

sp_sjobhistory

Description   Lists or deletes job history and job output logs. You can use this procedure to
                     look at the results of scheduled jobs that have completed. You can also use it
                     during administration of the job history and job output logs.

Syntax   sp_sjobhistory @name='...', @option='...'

Parameters   *name*
                     The name or ID of the scheduled job, a job, or a runid.

                     The @name argument can be used to restrict the scope to the history of a
                     single scheduled job, all scheduled jobs using a particular job, or a single run
                     of a scheduled job. By default, @name is the name or ID or a scheduled job.
                     To specify the name or ID of a job or as a runid, the @name argument is
                     prefixed with `jname` or `runid`.

                     option
                     Specifies a comma-separated list of options, and option_name=option_value
                     pairs that define the action to perform on a filtered set of the job history and
                     job output.

                     • list – lists the job history fields for the jobs matching the filter
                       conditions.

                     • list_short – lists a subset of the job history fields for the jobs matching
                       the filter conditions.

                     • drop – deletes the job history entries and job output entries for the jobs
                       matching the filter conditions.

                     • list_output – lists the job output for the jobs matching the filter
                       conditions.

- drop_output – deletes the job output entries for the jobs matching the filter conditions. The corresponding job history entries are not deleted.

- all_users – includes all user jobs in the scope. The caller must have the js_admin_role.

- user – restricts the scope to jobs run by or on behalf of this user. If the user specified is not the caller's server user name, the caller must have the js_admin_role.

- owner – restricts the scope to jobs run using this server user name. If the owner specified is not the caller's server user name, the caller must have the js_admin_role.

- age – restricts the scope to jobs recorded earlier than this number of days ago.

- minsize – restricts the scope to jobs with more than minsize bytes of output.

- force – allows the history and output to be deleted for running jobs.

Return value   Returns 0 on success or an error code.

Examples   **Example 1** The following example lists the concise history of the job called "orders_processed_report" that was run for the caller:

```
sp_sjobhistory @name='jname=orders_processed_report',
@option='list_short'
```

Example 2 The following example deletes the history and job output for the scheduled job that ran with runid 12389:

```
sp_sjobhistory @name='runid=12389', @option='drop'
```

Example 3 The following example lists all the caller's job history that has job output greater than 10,000 bytes:

```
sp_sjobhistory @option='list,minsize=10000'
```

Example 4 The following example drops all the caller job history and job output that has job output greater than 20,000 bytes:

```
sp_sjobhistory @option='drop,minsize=20000'
```

Example 5 The following example lists the history of the job called "load_sales_data" that was run by any user:

```
sp_sjobhistory @name='jname=load_sales_data',
@option='list,all_users'
```

Example 6 The following example lists the concise history of all the jobs run by the user called "mary":

```
sp_sjobhistory @option='list_short,user=mary'
```

Usage

- To protect against accidental removal of all the job history or output logs, an error occurs when you call sp_sjobhistory with the drop option and no filter arguments; you must supply at least one filter argument.

- sp_sjobhistory does not delete job output log entries for jobs that are currently running. These entries are silently ignored during the deletion process, unless the runid argument specifies an entry for a running job. In this case, sp_sjobhistory returns an error.

The following table shows the results from the js_history table when you use the list argument:

Table 5-2: Output returned from the list argument

Column	Type
job_runid	int
job_name	JS_DESC
job_state	char(2)
job_end	datetime
job_user_code	int
job_os_code	int
job_user_req	SUSER_NAME
job_long_message	JS_LMSG
sjob_id	JS_NAME_ID
sched_name	JS_NAME_ID
job_start	datetime
job_exit_code	int
job_atat_error	int
job_user_run	SUSER_NAME
job_short_message	JS_SMSG
job_size	int

The following table shows the results from the js_history table when you use the list_short argument:

Table 5-3: Output returned from the list_short argument

Column	Type
job_runid	int
job_name	JS_DESC
job_state	char(2)
job_end	datetime
job_user_code	int
job_size	int
sjob_id	JS_NAME_ID
sched_name	JS_NAME_ID
job_start	datetime
job_exit_code	int
job_atat_error	int

The following table shows the results from the js_history table when you use the list_output argument:

Table 5-4: Output returned from the list_output argument

Column	Type
job_runid	int
job_size	int
job_name	JS_DESC
job_output	JS_OUTPUT

Managing Job Scheduler in the Sybase Central ASE Plug-in

This chapter explains the procedures for using Job Scheduler in the ASE plug-in to the Sybase Central graphical management tool.

Note Before you can run scheduled jobs, you must perform the configuration procedures described in "Setting up Job Scheduler users" on page 9 and "Setting up access to target servers" on page 10.

Adding a scheduled job

❖ **Adding a scheduled job**

   1   From the appropriate Adaptive Server, select the Scheduled Jobs folder for which you want to manage your job.

   2   Double-click Add Scheduled Job in the right pane. The Scheduled Job Creation wizard opens.

3   Follow the steps in the wizard.

> **Note** When you create a name for a scheduled job, the name must begin with a letter. An error occurs if you create a name beginning with a digit.

> **Warning!** When you schedule your job, refer to the time on the server with your ASE Job Scheduler installation. The job will be run according to that server's time.

You can create a scheduled job from a Job Scheduler template. To use Job Scheduler templates you must first install them. See Chapter 3, "Using Templates to Create Jobs," for more information on templates.

Adding a job

❖   **Adding a job**

1   From the appropriate Adaptive Server, select the Scheduled Jobs folder to which you want to add your job.

2   Double-click the Jobs folder.

3   Double-click Add Job. The Job Creation wizard opens.

4   Follow the steps in the wizard.

> **Note** When you create a name for a job, the name must begin with a letter. An error occurs if you create a name beginning with a digit.

You can create a scheduled job from a Job Scheduler template. To use Job Scheduler templates you must first install them. See Chapter 3, "Using Templates to Create Jobs," for more information on templates.

Adding a schedule

❖ **Adding a schedule**

1   From the appropriate Adaptive Server, select the Scheduled Jobs folder to which you want to add a schedule.

2   Double-click the Schedules folder.

3   Double-click Add Schedule in the right pane. The Schedule Creation wizard opens.

4   Follow the steps in the wizard.

> **Note** When you create a name for a schedule, the name must begin with a letter. An error occurs if you create a name beginning with a digit.

Scheduling an existing job

❖ **Scheduling an existing job**

1   From the appropriate Adaptive Server, select the Scheduled Jobs folder to which you want to schedule an existing job.

2   Double-click the Jobs folder.

3   Select the Job you want to schedule.

4   Select File | Schedule. The Job Scheduler wizard opens.

5   Follow the steps in the wizard.

Viewing job history

❖ **Viewing the history of a scheduled job**

1   From the appropriate Adaptive Server, select the Scheduled Jobs folder from whose history you want to view.

2   Double-click the Job History folder.

3   View the history for your scheduled jobs.

4   Follow the steps in the wizard.

Purging job history

❖   **Purging the history of a job**

1   From the appropriate Adaptive Server, select the Scheduled Jobs folder from which you want to purge history.

2   Double-click the Job History folder.

3   Double-click Purge History. The Job History Purge wizard opens.

4   Follow the steps in the wizard.

Administering Job Scheduler

❖   **Administering Job Scheduler**

1   From the appropriate Adaptive Server, right-click the Scheduled Jobs folder for which you want to administer your job. The Job Scheduler Administration dialog box opens.

2   From the Task Configuration tab, you can:

   • See if the Job Scheduler Task is running.

   • Start or stop the Job Scheduler Task.

   • Set the Terminate jobs property. Then, you can set the property that specifies after how many seconds the termination occurs.

   • Set the property to enable Job Scheduler at start-up.

   • Set the Job Scheduler interval in minutes.

   • Set the maximum number of concurrent jobs.

   • Set the maximum job output size.

3   From the Database Configuration tab, you can:

- Define the percentage of the database to use for job history. When the clean task executes, old records are deleted until the amount of space defined by this property is left.

- Define the percentage of the database to use for output.

- Define the percentage of the space allocated for history that you want to keep free.

- Define the percentage of the space allocated for output that you want to keep free.

Showing all users

❖ **Showing all users**

1   From the appropriate Adaptive Server, select the Scheduled Jobs folder from which you want to show users.

2   Double-click the appropriate folder for the users you want to see:

   a   Double-click the Jobs folder. The jobs for all users appears in the right pane.

   b   Double-click the Schedules folder. The schedules for all users appears in the right pane.

   c   Double-click the Job History folder. The job history for all users appears in the right pane.

Managing a scheduled job

❖ **Managing a scheduled job**

1   From the appropriate Adaptive Server, select the Scheduled Jobs folder for which you want to manage a scheduled job.

2   Right-click the job ID of the job you want to manage. The scheduled job management submenu opens.

3   Choose one of the following the management options:

- Disable – temporarily disables a scheduled job. The entry is left in the Scheduled Jobs list with the job and schedule information intact but it is flagged as disabled. Subsequent scheduled executions will not be carried out until the job is re-enabled. Any current execution will continue until completion.

- Enable – enables a scheduled job.

- Run now – executes the job immediately, providing the job is not already running from its usual schedule**.**

- Terminate – If the job is executing, terminates the execution.

- Delete – deletes the job from the scheduled jobs.

Editing properties

❖ **Changing the properties of a job, schedule, or scheduled job**

1 From the appropriate Adaptive Server, select the Scheduled Jobs folder for which you want to edit properties.

2 Right-click on the scheduled job, job, or schedule for which you want to edit properties. The submenu opens.

3 Select Properties from the submenu. The Properties sheet for the object you selected opens.

The properties you can edit for a scheduled job are:

- Owner

- Target ASE

- Timeout in minutes

- Enable or disable log output

- Disable on failure

- Delete on completion

The properties you can edit for a schedule are:

- Name

- Owner

- Description

- Allow use by others

- Set start time

- Set start date

- Select no end date

- Set end date

- Set recurrence

> **Note** Changes to a schedule will affect all jobs using the same schedule.

The properties you can edit for a job are:

- Name

- Owner

- Description

- Selected template

- Job definition

- Allow multiple concurrent executions (defines whether subsequent executions of the job can commence if the last occurrence has not yet completed)

- Allow execution by others

- Allow execution only by job owner

- Timeout in minutes

Deleting a Job Scheduler object

❖ **Deleting a Job Scheduler object**

1 From the appropriate Adaptive Server, select the Scheduled Jobs folder and select the scheduled job, schedule, or job you want to delete.

2   From the menu bar, select Edit | Delete. The Job Scheduler object is
     deleted.

Note If a job or schedule is currently scheduled to run, it cannot be deleted.

CHAPTER 7   **Troubleshooting**

This chapter contains provides information about error logging and instructions for troubleshooting when creating and running scheduled jobs.

Topic	Page
Logging error messages	55
A job fails to run at the time you scheduled it	56
A scheduled job created from a template fails	56
A job with multiple calls to sp_sjobcmd fails	56
A stored procedure fails	57

Logging error messages

When you configure a new Adaptive Server, the installation program automatically sets the error log location. Then, Adaptive Server writes start-up information to the local error log file each time it starts. When you install Job Scheduler, the JS Agent creates its own log file in the same directory. See the configuration documentation for your platform to learn the default location and file name of the error log.

Many error messages from Adaptive Server only go to the user's terminal. However, fatal error messages (severity levels 19 and above), kernel error messages, and informational messages from Adaptive Server are recorded in the error log file.

Adaptive Server keeps the error log file open until you stop the server process. If you need to reduce the size of the error log by deleting old messages, stop the Adaptive Server process before you do so.

Note On some platforms (such as Windows NT), Adaptive Server also records error messages in the operating system event log. See the Adaptive Server installation and configuration guide for additional information about error logs.

A job fails to run at the time you scheduled it

When you create a schedule, or schedule a job, make sure the time you are using is the time on the server where ASE Job Scheduler is installed. If you schedule the job using the local time on your machine, it is not necessarily the same time as the time on the Job Scheduler installation server.

A scheduled job created from a template fails

If a scheduled job created from a template fails to run, check the job history for error messages. If a job failed to run, the reason why will be reported in the job history.

If you have not installed the stored procedures used by the templates on the target server where your job will run, you will see an error message similar to the following:

```
Procedure 'dump_databases' not found.  Specify
owner.objectname or use sp_help to check whether the
object exists (sp_help may produce lots of output).
```

See "Installing stored procedures on a target server" on page 14 for more information.

A job with multiple calls to sp_sjobcmd fails

When you create a job with several calls to sp_sjobcmd, you must have a new line after you execute the call:

```
go
'
```

A stored procedure fails

When Job Scheduler starts a job, it begins in the master database. If your job is invoking a stored procedure that is on another database, you may have to prefix the stored procedure with the name of the database on which it resides.

Adaptive Server Enterprise

Index

Symbols

T

U

V

W

X