System Administration Guide Volume 1

Adaptive Server Enterprise

12.5

DOCUMENT ID: 31654-01-1250-01

LAST REVISED: May 2001

Sybase, Inc., 6475 Christie Avenue, Emeryville, CA 94608.

Contents

About This Book

This manual, the *Sybase Adaptive Server System Administration Guide*, describes how to administer and control Sybase Adaptive Server Enterprise databases independent of any specific database application.

Audience

This manual is for Sybase System Administrators and Database Owners.

How to use this book

This manual contains the following chapters:

- Chapter 1, "Overview of System Administration," describes the structure of the Sybase system.
- Chapter 2, "System and Optional Databases," discusses the contents and function of the Adaptive Server system databases.
- Chapter 3, "System Administration for Beginners," summarizes important tasks that new System Administrators need to perform.
- Chapter 4, "Diagnosing System Problems," discusses Adaptive Server and Backup Server™ error handling and shows how to shut down servers and kill user processes.
- Chapter 5, "Setting Configuration Parameters," summarizes the configuration parameters that you set with sp_configure, which control many aspects of Adaptive Server behavior.
- Chapter 6, "Limiting Access to Server Resources," explains how to create and manage resource limits with Adaptive Server.

- Chapter 7, "Configuring Character Sets, Sort Orders, and Languages," discusses international issues, such as the files included in the Language Modules and how to configure an Adaptive Server language, sort order, and character set.
- Chapter 8, "Configuring Client/Server Character Set Conversions," discusses character set conversion between Adaptive Server and clients in a heterogeneous environment.
- Chapter 9, "Security Administration," provides an overview of the security features available in Adaptive Server.
- Chapter 10, "Managing Adaptive Server Logins and Database Users," describes methods for managing Adaptive Server login accounts and database users.
- Chapter 11, "Managing User Permissions," describes the use and implementation of user permissions.
- Chapter 12, "Auditing," describes how to set up auditing for your installation.
- Chapter 13, "Managing Remote Servers," discusses the steps the System Administrator and System Security Officer of each Adaptive Server must execute to enable remote procedure calls (RPCs).
- Chapter 14, "Using Kerberos, DCE, and Windows NT LAN Manager," describes the network-based security services that enable you to authenticate users and protect data transmitted among machines on a network.
- Chapter 15, "Overview of Disk Resource Issues," provides an overview of Adaptive Server disk resource issues.
- Chapter 16, "Initializing Database Devices," describes how to initialize and use database devices.
- Chapter 17, "Mirroring Database Devices," describes how to mirror database devices for nonstop recovery from media failures.
- Chapter 18, "Configuring Memory," explains how to configure Adaptive Server to use the available memory on your system.
- Chapter 19, "Configuring Data Caches," discusses how to create named caches in memory and bind objects to those caches.
- Chapter 20, "Managing Multiprocessor Servers," explains how to use multiple CPUs with Adaptive Server and discusses system administration issues that are unique to symmetric multiprocessing (SMP) environments.

- Chapter 21, "Creating and Managing User Databases," discusses the physical placement of databases, tables, and indexes, and the allocation of space to them.
- Chapter 22, "Setting Database Options," describes how to set database options.
- Chapter 23, "Creating and Using Segments," describes how to use segments, which are named collections of database devices, in databases.
- Chapter 24, "Using the reorg Command," describes how to use the reorg command.
- Chapter 25, "Checking Database Consistency," describes how to use the database consistency checker, dbcc, to detect and fix database problems.
- Chapter 26, "Developing a Backup and Recovery Plan," discusses the capabilities of the Backup Server and how to develop your backup strategy.
- Chapter 27, "Backing Up and Restoring User Databases," discusses how to recover user databases.
- Chapter 28, "Restoring the System Databases," discusses how to recover system databases.
- Chapter 29, "Managing Free Space with Thresholds," discusses managing space with thresholds.

Related documents

The following documents comprise the Sybase Adaptive Server Enterprise documentation:

- 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.
- *Configuring Adaptive Server Enterprise* for your platform – provides instructions for performing specific configuration tasks for Adaptive Server.

- *What's New in Adaptive Server Enterprise?* – describes the new features in Adaptive Server version 12.5, the system changes added to support those features, and the changes that may affect your existing applications.
- *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. This manual also contains descriptions of the pubs2 and pubs3 sample databases.
- *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.
- *Reference Manual* – contains detailed information about all Transact-SQL commands, functions, procedures, and datatypes. This manual also contains a list of the Transact-SQL reserved words and definitions of system tables.
- *Performance and Tuning Guide* – explains how to tune Adaptive Server for maximum performance. This manual includes information about database design issues that affect performance, query optimization, how to tune Adaptive Server for very large databases, disk and cache issues, and the effects of locking and cursors on performance.
- The *Utility Guide* – documents the Adaptive Server utility programs, such as isql and bcp, which are executed at the operating system level.
- The *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. Available only in print version.
- The *System Tables Diagram* – illustrates system tables and their entity relationships in a poster format. Available only in print version.
- *Error Messages and Troubleshooting Guide* – explains how to resolve frequently occurring error messages and describes solutions to system problems frequently encountered by users.
- *Component Integration Services User's Guide* – explains how to use the Adaptive Server Component Integration Services feature to connect remote Sybase and non-Sybase databases.

- *Java in Adaptive Server Enterprise* – describes how to install and use Java classes as datatypes, functions, and stored procedures in the Adaptive Server database.
- *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.
- *Using Adaptive Server Distributed Transaction Management Features* – explains how to configure, use, and troubleshoot Adaptive Server DTM features in distributed transaction processing environments.
- *EJB Server User's Guide* – explains how to use EJB Server to deploy and execute Enterprise JavaBeans in Adaptive Server.
- *XA Interface Integration Guide for CICS, Encina, and TUXEDO* – provides instructions for using Sybase's DTM XA interface with X/Open XA transaction managers.
- *Glossary* – defines technical terms used in the Adaptive Server documentation.
- *Sybase jConnect for JDBC Programmer's Reference* – describes the jConnect for JDBC product and explains how to use it to access data stored in relational database management systems.
- *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.
- *Historical Server User's Guide* –describes how to use Historical Server to obtain performance information for SQL Server and Adaptive Server.
- *Monitor Server User's Guide* – describes how to use Monitor Server to obtain performance statistics from SQL Server and Adaptive Server.
- *Monitor Client Library Programmer's Guide* – describes how to write Monitor Client Library applications that access Adaptive Server performance data.

Other sources of information

Use the Sybase Technical Library CD and the Technical Library Product Manuals Web site to learn more about your product:

- Technical Library CD contains product manuals and is included with your software. The DynaText browser (downloadable from Product Manuals at http://www.sybase.com/detail/1,3693,1010661,00.html) allows you to access technical information about your product in an easy-to-use format.

Refer to the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting the Technical Library.

- Technical Library Product Manuals Web site is an HTML version of the Technical Library CD that you can access using a standard Web browser. In addition to product manuals, you will find links to the Technical Documents Web site (formerly known as Tech Info Library), the Solved Cases page, and Sybase/Powersoft newsgroups.

  To access the Technical Library 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.

v **For 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.

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

5 Click a Certification Report title to display the report.

v **For the latest information on EBFs and Updates**

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

2 Select EBFs/Updates. Enter user name and password information, if prompted (for existing Web accounts) or create a new account (a free service).

3 Specify a time frame and click Go.

4 Select a product.

5 Click an EBF/Update title to display the report.

v **To create 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.

Conventions used in this manual

This section describes the style conventions used in this manual.

Formatting SQL statements

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 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.

SQL syntax conventions

Table 1 lists the conventions for syntax statements in this manual:

Table 1: Syntax statement conventions

Key	Definition
`command`	Command names, command option names, utility names, utility flags, and other keywords are in `bold Courier` in syntax statements, and in bold Helvetica in paragraph text.
`variable`	Variables, or words that stand for values that you fill in, are in italics.
`{ }`	Curly braces indicate that you choose at least one of the enclosed options. Do not include braces in your option.
`[ ]`	Square brackets mean choosing one or more of the enclosed options is optional. Do not include brackets in your option.
`( )`	Type parentheses as part of the command.
`|`	The vertical bar means you may select only one of the options shown.
`,`	The comma means you may choose as many of the options shown as you like, separating your choices with commas.

- Syntax statements (displaying the syntax and all options for a command) are printed like this:

  ```
  sp_dropdevice [device_name]
  ```

  or, 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: normal font for keywords, italics for user-supplied words.

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

  ```
  select * from publishers
  ```

- Examples of output from the computer are printed like this:

```
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)
```

Case

You can disregard case when you type keywords:

SELECT is the same as Select is the same as select.

Obligatory options {you must choose at least one}

- *Curly braces and vertical bars:* Choose *one and only one* option.

```
{die_on_your_feet | live_on_your_knees | live_on_your_feet}
```

- *Curly braces and commas:* Choose *one or more* options. If you choose more than one, separate your choices with commas.

```
{cash, check, credit}
```

Optional options

- *One item in square brackets:* You don't have to choose it.

```
[anchovies]
```

- *Square brackets and vertical bars:* Choose *none or only one*.

```
[beans | rice | sweet_potatoes]
```

- *Square brackets and commas:* Choose *none, one, or more than one* option. If you choose more than one, separate your choices with commas.

```
[extra_cheese, avocados, sour_cream]
```

Ellipsis

An ellipsis (. . .) means that you can *repeat* the last unit as many times as you like. In this syntax statement, buy is a required keyword:

```
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.

An ellipsis can also be used inline to signify portions of a command that are left out of a text example. The following syntax statement represents the complete create database command, even though required keywords and other options are missing:

```
create database...for load
```

Expressions

Several different types of **expressions** are used in Adaptive Server syntax statements.

Table 2: Types of expressions used in syntax statements

Usage	Definition
expression	Can include constants, literals, functions, column identifiers, variables, or parameters
logical_expression	An expression that returns TRUE, FALSE, or UNKNOWN
constant_expression	An expression that always returns the same value, such as “5+3” or “ABCDE”
float_expr	Any floating-point expression or expression that implicitly converts to a floating value
integer_expr	Any integer expression or an expression that implicitly converts to an integer value
numeric_expr	Any numeric expression that returns a single value
char_expr	An expression that returns a single character-type value
binary_expression	An expression that returns a single binary or varbinary value

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 of System Administration

This chapter introduces the basic topics of Adaptive Server system administration, including:

Adaptive Server administration tasks

Administering Adaptive Server includes tasks such as:

- Installing Adaptive Server and Backup Server
- Creating and managing Adaptive Server login accounts
- Granting roles and permissions to Adaptive Server users
- Managing and monitoring the use of disk space, memory, and connections
- Backing up and restoring databases
- Diagnosing system problems
- Configuring Adaptive Server to achieve the best performance

In addition, System Administrators may have a hand in certain database design tasks, such as enforcing integrity standards. This function may overlap with the work of application designers.

Although a System Administrator concentrates on tasks that are independent of the applications running on Adaptive Server, he or she is likely to be the person with the best overview of all the applications. For this reason, a System Administrator can advise application designers about the data that already exists on Adaptive Server, make recommendations about standardizing data definitions across applications, and so on.

However, the distinction between what is specific to an application is sometimes a bit "fuzzy." Owners of user databases will consult certain sections of this book. Similarly, System Administrators and Database Owners will use the *Transact-SQL User's Guide* (especially the chapters on data definition, stored procedures, and triggers). Both System Administrators and application designers will use the *Performance and Tuning Guide*.

Roles required for system administration tasks

Many of the commands and procedures discussed in this manual require the System Administrator or System Security Officer role. Other sections in this manual are relevant to Database Owners. A Database Owner's user name within the database is "dbo". You cannot log in as "dbo:" a Database Owner logs in under his or her Adaptive Server login name and is recognized as "dbo" by Adaptive Server only while he or she is using the database.

Various security-related, administrative, and operational tasks are grouped into the following system roles:

- **System Administrator**, whose tasks include:
  - Managing disk storage
  - Monitoring Adaptive Server's automatic recovery procedure
  - Fine-tuning Adaptive Server by changing configurable system parameters
  - Diagnosing and reporting system problems
  - Backing up and loading databases
  - Granting and revoking the System Administrator role
  - Modifying and dropping server login accounts
  - Granting permissions to Adaptive Server users

    - Creating user databases and granting ownership of them
    - Setting up groups which can be used for granting and revoking permissions)
- **System Security Officer**, who performs security-related tasks such as:
    - Creating server login accounts, which includes assigning initial passwords
    - Changing the password of any account
    - Granting and revoking the System Security Officer and Operator roles
    - Creating, granting, and revoking user-defined roles
    - Granting the capability to impersonate another user throughout the server
    - Setting the password expiration interval
    - Setting up Adaptive Server to use network-based security services
    - Managing the audit system
- **Operator**, a user who can back up and load databases on a server-wide basis. The operator role allows a single user to use the dump database, dump transaction, load database, and load transaction commands to back up and restore all databases on a server without having to be the owner of each one. These operations can be performed in a single database by the Database Owner or a System Administrator.

These roles provide individual accountability for users performing operational and administrative tasks. Their actions can be audited and attributed to them. A System Administrator operates outside the discretionary access control (DAC) protection system; that is, when a System Administrator accesses objects Adaptive Server does not check the DAC permissions.

In addition, two kinds of object owners have special status because of the objects they own. These ownership types are:

- Database Owner
- Database object owner

Database Owner

The **Database Owner** is the creator of a database or someone to whom database ownership has been transferred. A System Administrator grants users the authority to create databases with the grant command.

A Database Owner logs in to Adaptive Server using his or her assigned login name and password. In other databases, that owner is known by his or her regular user name. In the database Adaptive Server recognizes the user as having the "dbo" account.

A Database Owner can:

- Run the system procedure sp_adduser to allow other Adaptive Server users access to the database
- Use the grant command to give other users permission to create objects and execute commands within the database

Adding users to databases is discussed in Chapter 10, "Managing Adaptive Server Logins and Database Users." Granting permissions to users is discussed in Chapter 11, "Managing User Permissions."

The Database Owner does not automatically receive permissions on objects owned by other users. However, a Database Owner can temporarily assume the permissions of other users in the database at any time by using the setuser command. Using a combination of the setuser and grant commands, the Database Owner can acquire permissions on any object in the database.

Note Because the Database Owner role is so powerful, the System Administrator should plan carefully who should own databases in the server. The System Security Officer should consider auditing the database activity of all Database Owners.

Database object owner

A **Database object owner** is a user who creates a database object. **Database object**s are tables, indexes, views, defaults, triggers, rules, constraints, and procedures. Before a user can create a database object, the Database Owner must grant the user permission to create objects of a particular type. There is no special login name or password for a database object owner.

The database object owner creates an object using the appropriate create statement, and then grants permission to other users.

The creator of a database object is automatically granted all permissions on that object. The System Administrator also has all permissions on the object. The owner of an object must explicitly grant permissions to other users before they can access the object. Even the Database Owner cannot use an object directly unless the object owner grants him or her the appropriate permission. However, the Database Owner can always use the setuser command to impersonate any other user in the database, including the object owner.

Note When a database object is owned by someone other than the Database Owner, the user (including a System Administrator) must qualify the name of that object with the object owner's name—*ownername*.*objectname*—to access the object. If an object or a procedure needs to be accessed by a large number of users, particularly in ad hoc queries, having these objects owned by "dbo" greatly simplifies access.

Using *isql* to perform system administration tasks

This book assumes that you will perform the system administration tasks described in this guide by using the command-line utility isql. This section provides some basic information about using isql. For complete information about isql, see the *Utilites Guide*.

You can also use the graphic tool Sybase Central™ to perform many of the tasks described in this book, as described in "Using Sybase Central for system administration tasks" on page 7.

Starting *isql*

To start isql on most platforms, type this command at an operating system prompt:

```
isql -Uusername
```

where *username* is the user name of the System Administrator. Adaptive Server prompts you for your password.

Note Do not use the -P option of isql to specify your password because another user might then see your password.

You can use isql in command-line mode to enter many of the Transact-SQL examples in this manual.

Entering statements

The statements that you enter in isql can span several lines. isql does not process statements until you type "go" on a separate line. For example:

```
1> select *
2> from sysobjects
3> where type = "TR"
4> go
```

The examples in this manual do not include the go command between statements. If you are typing the examples, you must enter the go command to see the sample output.

Saving and reusing statements

This manual frequently suggests you that save the Transact-SQL statements you use to create or modify user databases and database objects. The easiest way to do this is to create or copy the statements to an ASCII-formatted file. You can then use the file to supply statements to isql if you need to re-create databases or database objects later.

The syntax for using isql with an ASCII-formatted file is:

```
isql -Uusername -ifilename
```

where *filename* is the full path and file name of the file that contains Transact-SQL statements. On UNIX and other platforms, use the less than symbol (<) to redirect the file.

The Transact-SQL statements in the ASCII file must use valid syntax and the go command.

Using Sybase Central for system administration tasks

You can accomplish many of the system administration tasks detailed in this book with Sybase Central, a graphic tool that comes with Adaptive Server.

Here are some of the tasks you can use Sybase Central for:

- Initializing database devices (Windows NT servers only)
- Setting configuration parameters
- Viewing the amount of free log space in a database
- Generating data definition language (DDL)
- Creating logins
- Adding remote servers
- Creating databases
- Creating stored procedures
- Defining roles
- Adding data caches
- Setting database options
- Backing up and restoring databases

You can also use the Monitor Viewer feature of Sybase Central to access Adaptive Server Monitor™. Sybase Central also comes with extensive online help.

You can use the Sybase Central DDL-generation feature to record your work to Transact-SQL scripts. The DDL-generation feature lets you save to a script the actions you performed in an entire server or within a specific database.

System tables

The master database contains **system table**s that keep track of information about Adaptive Server as a whole. In addition, each database (including the master database) contains system tables that keep track of information specific to that database.

All the Adaptive Server-supplied tables in the master database (Adaptive Server's controlling database) are considered system tables. Each user database is created with a subset of these system tables. The system tables may also be referred to as the **data dictionary** or the system catalogs.

A master database and its tables are created when Adaptive Server is installed. The system tables in a user database are created when the create database command is issued. The names of all system tables start with "sys". You cannot create tables in user databases that have the same names as system tables. An explanation of the system tables and their columns is included in the *Adaptive Server Reference Manual*.

Querying the system tables

You can query system tables just like any other tables. For example, the following statement returns the names of all the triggers in the database:

```
select name
from sysobjects
where type = "TR"
```

In addition, Adaptive Server supplies **stored procedure**s (called **system procedures**), many of which provide shortcuts for querying the system tables.

Here are the system procedures that provide information from the system tables:

• sp_commonkey	• sp_helpremotelogin
• sp_configure	• sp_help_resource_limit
• sp_countmedatada	• sp_helprotect
• sp_dboption	• sp_helpsegment
• sp_estspace	• sp_helpserver
• sp_help	• sp_helpsort
• sp_helpartition	• sp_helptext
• sp_helpcache	• sp_helpthreshold
• sp_helpconfig	• sp_helpuser
• sp_helpconstraint	• sp_lock
• sp_helpdb	• sp_monitor
• sp_helpdevice	• sp_monitorconfig
• sp_helpgroup	• sp_procqmode
• sp_helpindex	• sp_showcontrolinfo

• sp_helpjava	• sp_showexeclass
• sp_helpjoins	• sp_showplan
• sp_helpkey	• sp_spaceused
• sp_helplanguage	• sp_who
• sp_helplog	• sp_help_resource_limit

For complete information about the system procedures, see the *Adaptive Server Reference Manual*.

Keys in system tables

Primary, foreign, and common keys for the system tables are defined in the master and model databases. You can get a report on defined keys by executing sp_helpkey. For a report on columns in two system tables that are likely join candidates, execute sp_helpjoins.

The *Adaptive Server System Tables Diagram* included with Adaptive Server shows the relationships between columns in the system tables.

Updating system tables

The Adaptive Server system tables contain information that is critical to the operation of your databases. Under ordinary circumstances, you do not need to perform direct data modifications to system tables.

Update system tables only when you are instructed to do so by Sybase Technical Support or by an instruction in the *Troubleshooting Guide* or in this manual.

When you update system tables, you must issue an sp_configure command that enables system table updates. While this command is in effect, any user with appropriate permission can modify a system table. Other requirements for direct changes to system tables are:

- Modify system tables only inside a transaction. Issue a begin transaction command before you issue the data modification command.
- Verify see that only the rows you wanted changed were affected by the command and that the data was changed correctly.

- If the command was incorrect, issue a rollback transaction command. If the command was correct, issue a commit transaction command.

Warning! Some system tables should not be altered by any user under any circumstances. Some system tables are built dynamically by system processes, contain encoded information, or display only a portion of their data when queried. Imprudent, ad hoc updates to certain system tables can make Adaptive Server unable to run, make database objects inaccessible, scramble permissions on objects, or terminate a user session.
Moreover, you should never attempt to alter the definition of the system tables in any way. For example, do not alter system tables to include constraints. Triggers, defaults, and rules are not allowed in system tables. If you try to create a trigger or bind a rule or default to a system table, you will get an error message.

System procedures

The names of all system procedures begin with "sp_". They are located in the sybsystemprocs database, but you can run many of them in any database by issuing the stored procedure from the database or by qualifying the procedure name with the database name.

If you execute a system procedure in a database other than sybsystemprocs, it operates on the system tables in the database from which it was executed. For example, if the Database Owner of pubs2 runs sp_adduser from pubs2 or issues the command pubs2..sp_adduser, the new user is added to pubs2..sysusers. However, this does not apply to system procedures that update only tables in the master database.

Permissions on system procedures are discussed in the *Adaptive Server Reference Manual*.

Using system procedures

A **parameter** is an argument to a stored or system procedure. If a parameter value for a system procedure contains reserved words, punctuation, or embedded blanks, it must be enclosed in single or double quotes. If the parameter is an object name, and the object name is qualified by a database name or owner name, the entire name must be enclosed in single or double quotes.

System procedures can be invoked by sessions using either chained or unchained transaction mode. However, the system procedures that modify data in system tables in the master database cannot be executed from within a transaction, since this could compromise recovery. The system procedures that create temporary work tables cannot be run from transactions.

If no transaction is active when you execute a system procedure, Adaptive Server turns off chained mode and sets transaction isolation level 1 for the duration of the procedure. Before returning, the session's chained mode and isolation level are reset to their original settings. For more information about transaction modes and isolation levels, see the *Adaptive Server Reference Manual*.

All system procedures report a return status. For example:

```
return status = 0
```

means that the procedure executed successfully.

System procedure tables

The system procedures use several *system procedure tables* in the master and sybsystemdb databases to convert internal system values (for example, status bits) into human-readable format. One of these tables, spt_values, is used by a variety of system procedures, including:

• sp_configure	• sp_helpdevice
• sp_dboption	• sp_helpindex
• sp_depends	• sp_helpkey
• sp_help	• sp_helprotect
• sp_helpdb	• sp_lock

The spt_values table can be updated only by an upgrade; it cannot be modified otherwise. To see how it is used, execute sp_helptext and look at the text for one of the system procedures that references it.

The other system procedure tables are spt_monitor, spt_committab, and tables needed by the catalog stored procedures. (The spt_committab table is located in the sybsystemdb database.)

In addition, several of the system procedures create and then drop temporary tables. For example, sp_helpdb creates #spdbdesc, sp_helpdevice creates #spdevtab, and sp_helpindex creates #spindtab.

Creating system procedures

Many of the system procedures are explained in this manual, in the sections where they are relevant. For complete information about system procedures, see the *Adaptive Server Reference Manual*.

System Administrators can write system procedures that can be executed in any database. Simply create a stored procedure in sybsystemprocs and give it a name that begins with "sp_". The uid of the stored procedure must be 1, the uid of the Database Owner.

Most of the system procedures that you create query the system tables. You can also create stored procedures that modify the system tables, although this is not recommended.

To create a stored procedure that modifies system tables, a System Security Officer must first turn on the allow updates to system tables configuration parameter. Any stored procedure created while this parameter is set to "on" will *always* be able to update system tables, even when allow updates to system tables is set to "off." To create a stored procedure that updates the system tables:

1 Use sp_configure to set allow updates to system tables to "on."

2 Create the stored procedure with the create procedure command.

3 Use sp_configure to set allow updates to system tables to "off."

Warning! Use extreme caution when you modify system tables. Always test the procedures that modify system tables in development or test databases, not in your production database.

System extended stored procedures

An extended stored procedure (ESP) provides a way to call external language functions from within Adaptive Server. Adaptive Server provides a set of ESPs; users can also create their own. The names of all system extended stored procedures begin with "xp_", and are located in the sybsystemprocs database.

One very useful system ESP is xp_cmdshell, which executes an operating system command on the system that is running Adaptive Server.

You can invoke a system ESP just like a system procedure. The difference is that a system ESP executes procedural language code rather than Transact-SQL statements. All ESPs are implemented by an Open Server application called XP Server, which runs on the same machine as Adaptive Server. XP Server starts automatically on the first ESP innovation.

For information about the system ESPs provided with Adaptive Server, see the *Adaptive Server Reference Manual.*

Creating system ESPs

Create a system ESP in the sybsystemprocs database using the create procedure command. System procedures are automatically included in the sybsystemprocs database. The name of the ESP, and its procedural language function, should begin with "xp_". The uid of the stored procedure must be 1, the uid of the Database Owner.

For general information about creating ESPs, see Chapter 15, "Using Extended Stored Procedures"in the *Transact-SQL User's Guide*.

Logging error messages

Adaptive Server writes start-up information to a local error log file each time it boots. The installation program automatically sets the error log location when you configure a new Adaptive Server. 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 go to the user's terminal only. 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.

Connecting to Adaptive Server*

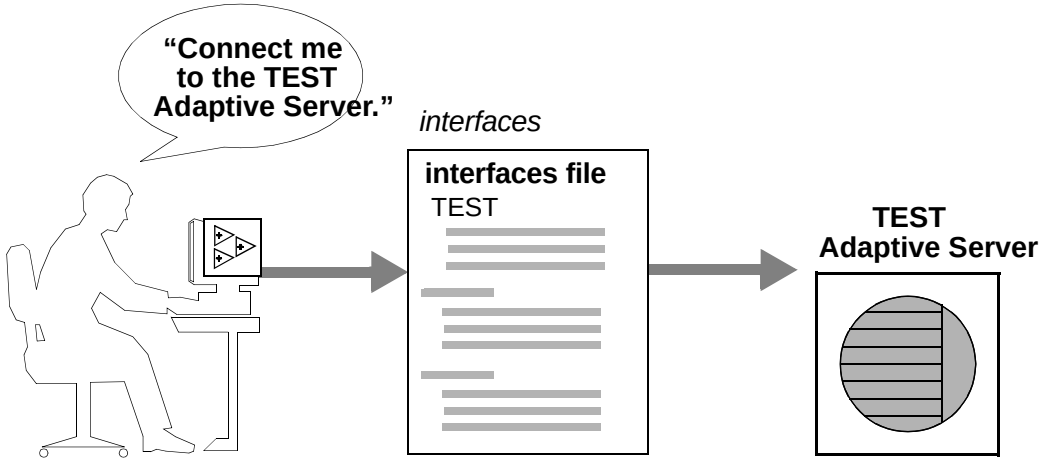
Adaptive Server can communicate with other Adaptive Servers, Open Server applications, and client software on the network. Clients can talk to one or more servers, and servers can communicate with other servers via remote procedure calls. In order for products to interact with one another, each needs to know where the others reside on the network. This network service information is stored in the interfaces file.

The interfaces file

The interfaces file is usually named *interfaces*, *interfac*, or *sql.ini*, depending on the operating system.

The interfaces file is like an address book. It lists the name and address of every known server. When you use a client program to connect to a server, the program looks up the server name in the interfaces file and then connects to the server using the address, as shown in Figure 1-1.

Figure 1-1: Connecting to Adaptive Server



The name, location, and contents of the interfaces file differ between operating systems. Also, the format of the Adaptive Server addresses in the interfaces file differs between network protocols.

When you install Adaptive Server, the installation program creates a simple interfaces file that you can use for local connections to Adaptive Server over one or more network protocols. As a System Administrator, it is your responsibility to modify the interfaces file and distribute it to users so that they can connect to Adaptive Server over the network. See the configuration documentation for your platform for information about the interfaces file for your platform.

Directory services

A directory service manages the creation, modification, and retrieval of network service information. Directory services are provided by platform or third-party vendors and must be purchased and installed separately from Adaptive Server. Two examples of directory services are NT Registry and Distributed Computing Environment (DCE).

The *$SYBASE/config/libtcl.cfg* file is a Sybase-supplied configuration file used by servers and clients to determine:

- Which directory service to use, and

- The location of the specified directory service driver.

If no directory services are installed or listed in the *libtcl.cfg* file, Adaptive Server defaults to the interfaces file for obtaining network service information.

The System Administrator must modify the *libtcl.cfg* file as appropriate for the operating environment.

Some directory services are specific to a given platform; others can be used on several different platforms. Because of the platform-specific nature of directory services, refer to the configuration documentation for your platform for detailed information on configuring for directory services.

LDAP as a directory service

Lightweight Directory Access Protocol (LDAP) is an industry standard for accessing directory services. Directory services allow components to look up information by a distinguished name (DN) from an LDAP server that stores and manages server, user, and software information that is used throughout the enterprise or over a network.

The LDAP server can be located on a different platform from the one on which Adaptive Server or the clients are running. LDAP defines the communication protocol and the contents of messages exchanged between clients and servers. Messages are operators, such as client requests for read, write and query, and server responses, including data-format information.

The LDAP server stores and retrieves information about:

- Adaptive Server, such as IP address, port number, and network protocol
- Security mechanisms and filters
- High availability companion server name

The LDAP server can be configured with these access restrictions:

- Anonymous authentication – all data is visible to any user.
- User name and password authentication – Adaptive Server uses the default user name and password from the file:

  UNIX, 32-bit – *$SYBASE/$SYBASE_OCS/config/libtcl.cfg*

UNIX, 64-bit – *$SYBASE/$SYBASE_OCS/config/libtcl64.cfg*

NT – *%SYBASE%\%SYBASE_OCS%\ini\libtcl.cfg*

User name and password authentication properties establish and end a session connection to an LDAP server.

Note The user name and password that are passed to the LDAP server for user authentication purposes are distinct and different from those used to access Adaptive Server.

When an LDAP server is specified in the *libtcl.cfg* or *libtcl64.cfg* file (collectivley *libtcl*.cfg* file), the server information is accessible only from the LDAP server. Adaptive Server ignores the interfaces file.

If multiple directory services are supported in a server, then the order in which they are searched is specified in *libtcl*.cfg*. You cannot specify the search order with the dataserver command-line option.

Multiple directory services

Any type of LDAP service, whether it is an actual server or a gateway to other LDAP services, is called an LDAP server.

You can specify multiple directory services for high-availability failover protection in *libtcl*.cfg*. Not every directory service in the list needs to be an LDAP server.

For example:

```
[DIRECTORY]
   ldap=libdldap.so ldap://test:389/dc=sybase,dc=com
   dce=libddce.so ditbase=/.:/subsys/sybase/dataservers
   ldap=libdldap.so ldap://huey:11389/dc=sybase,dc=com
```

In this example, if the connection to *test:389* fails, the connection fails over to the DCE driver with the specified DIT base. If this also fails, a connection to the LDAP server on *huey:11389* is attempted. Different vendors employ different DIT base formats.

Note For more information, see the *Open Client Client-Library/C Programmer's Guide* and the *Open Client Client-Library/C Reference Manual*.

LDAP directory services versus the Sybase interfaces file

The LDAP driver implements directory services for use with an LDAP server. LDAP directories are an infrastructure that provide:

- A network-based alternative to the traditional Sybase interfaces file
- A single, hierarchical view of information, including users, software, resources, networks, files, and so on

Table 1-1 highlights the differences between the Sybase interfaces file and an LDAP server.

Table 1-1: interfaces file versus LDAP directory services

interfaces file	Directory services
Platform-specific	Platform-independent
Specific to each Sybase installation	Centralized and hierarchical
Contains separate master and query entries	One entry for each server that is accessed by both clients and servers
Cannot store metadata about the server	Stores metadata about the server

Performance

Performance when using an LDAP server may be slower than when using an interfaces file because the LDAP server requires time to make a network connection and retrieve data. Since this connection is made when Adaptive Server is started, changes in performance will be seen at login time, if at all. During normal system load, the delay should not be noticeable. During high system load with many connections, especially repeated connections with short duration, the overall performance difference of using an LDAP server versus the traditional interfaces file might be noticeable.

Security features available in Adaptive Server

SQL Server release 11.0.6 passed the security evaluation by the National Security Agency (NSA) at the Class C2 criteria. (The requirements for the C2 criteria are given by the Department of Defense in DOD 52.00.28-STD, *Department of Defense Trusted Computer System Evaluation Criteria* [TCSEC], also known as the “Orange Book.”)

The configuration of SQL Server release 11.0.6 that was evaluated at the C2 security level by the NSA in 1996 on the HP 9000 HP-UX BLS, 9.09+ platform is referred to as the **evaluated configuration**. Certain features of SQL Server, such as remote procedures and direct updates to system tables, were excluded from the evaluated configuration. Notes in the Adaptive Server documentation indicate particular features that were not included in the evaluated configuration. For a complete list of features that were excluded from the evaluated configuration, see Appendix A in the *SQL Server Installation and Configuration Guide for HP 9000 HP-UX BLS, 9.09+*.

This version of Adaptive Server contains all of the security features included in SQL Server release 11.0.6 plus some new security features. Table 1-2 summarizes the major features.

Table 1-2: Major security features

Security feature	Description
Discretionary Access Controls (DAC)	Provides access controls that give object owners the ability to restrict access to objects, usually with the grant and revoke commands. This type of control is dependent upon an object owner's discretion.
Identification and authentication controls	Ensures that only authorized users can log in to the system.
Division of roles	Allows you to grant privileged roles to specified users so that only designated users can perform certain tasks. Adaptive Server has predefined roles, called "system roles," such as System Administrator and System Security Officer. In addition, Adaptive Server allows System Security Officers to define additional roles, called "user-defined roles."
Network-based security	Provides security services to authenticate users and protect data transmitted among machines on a network.
Auditing	Provides the capability to audit events such as logins, logouts, server boot operations, remote procedure calls, accesses to database objects, and all actions by a specific user or with a particular role active. In addition, Adaptive Server provides a single option to audit a set of server-wide security-relevant events.

CHAPTER 2

System and Optional Databases

This chapter describes the system databases that reside on all Adaptive Server systems. It also describes optional Sybase-supplied databases that you can install, and a database that Sybase Technical Support may install for diagnostic purposes.

Topics covered in this chapter include:

Overview of system databases

When you install Adaptive Server, it includes these system databases:

- The master database
- The model database
- The system procedure database, sybsystemprocs
- The temporary database, tempdb

Optionally, you can install:

- The auditing database, sybsecurity
- The two-phase commit transaction database, sybsystemdb

- The sample databases, pubs2 and pubs3
- The dbcc database, dbccdb

For information about installing the master, model, sybsystemprocs, and tempdb databases, see the installation documentation for your platform. For information on installing dbccdb, see Chapter 25, "Checking Database Consistency."

The master, model, and temporary databases reside on the device named during installation, which is known as the master device. The master database is contained entirely on the master device and cannot be expanded onto any other device. All other databases and user objects should be created on other devices.

Warning! Do not store user databases on the master device. Storing user databases on the master device makes it difficult to recover the system databases if they become damaged. Also, you will not be able to recover user databases stored on the master device.

You should install the sybsecurity and sybsystemdb databases on their own devices and segment. For more information, see the installation documentation for your platform.

You can install the sybsystemprocs database on a device of your choice. You may want to modify the installation scripts for pubs2 and pubs3 to share the device you create for sybsystemprocs.

The *installpubs2* and the *installpubs3* scripts do not specify a device in their create database statement, so they are created on the default device. At installation time, the master device is the default device. To change this, you can either edit the scripts or follow the instructions in Chapter 16, "Initializing Database Devices," for information about adding more database devices and designating default devices.

master database

The master database controls the operation of Adaptive Server and stores information about all user databases and their associated database devices. Table 2-1 describes information the master database tracks.

Table 2-1: Information the master database tracks

Information	System Table
User accounts	syslogins
Remote user accounts	sysremotelogins
Remote servers that this server can interact with	sysservers
Ongoing processes	sysprocesses
Configurable environment variables	sysconfigures
System error messages	sysmessages
Databases on Adaptive Server	sysdatabases
Storage space allocated to each database	sysusages
Tapes and disks mounted on the system	sysdevices
Active locks	syslocks
Character sets	syscharsets
Languages	syslanguages
Users who hold server-wide roles	sysloginroles
Server roles	syssrvroles
Adaptive Server engines that are online	sysengines

Because the master database stores information about user databases and devices, you must be in the master database in order to issue the create database, alter database, disk init, disk refit, disk reinit, and disk mirroring commands.

Controlling object creation in *master*

When you first install Adaptive Server, only a System Administrator can create objects in the master database, because the System Administrator implicitly becomes "dbo" of any database he or she uses. Any objects created on the master database should be used for the administration of the system as a whole. Permissions in master should remain set so that most users cannot create objects there.

Warning! Never place user objects in master. Storing user objects in master can cause the transaction log to fill quickly. If the transaction log runs out of space completely, you will not be able to use dump transaction commands to free space in master.

Another way to discourage users from creating objects in master is to change the default database for users (the database to which a user is connected when he or she logs in) with sp_modifylogin. See Chapter , "Adding users to databases," for more information.

If you create your own system procedures, create them in the sybsystemprocs database rather than in master.

Backing up *master* and keeping copies of system tables

To be prepared for hardware or software failure on Adaptive Server, the two most important housekeeping tasks are:

- Performing frequent backups of the master database and all user databases. See "Keep up-to-date backups of master" on page 38 for more information. See also Chapter 28, "Restoring the System Databases," for an overview of the process for recovering the master database.
- Keeping a copy (preferably offline) of these system tables: sysusages, sysdatabases, sysdevices, sysloginroles, and syslogins. See "Keep offline copies of system tables" on page 39 for more information. If you have copies of these scripts, and a hard disk crash or other disaster makes your database unusable, you can use the recovery procedures described in Chapter 28, "Restoring the System Databases." If you do not have current copies of your scripts, it will be much more difficult to recover Adaptive Server when the master database is damaged.

model database

Adaptive Server includes the model database, which provides a template, or prototype, for new user databases. Each time a user enters the create database command, Adaptive Server makes a copy of the model database and extends the new database to the size specified by the create database command.

Note A new database cannot be smaller than the model database.

The model database contains the required system tables for each user database. You can modify model to customize the structure of newly created databases—everything you do to model will be reflected in each new database. Some of the changes that System Administrators commonly make to model are:

- Adding user-defined datatypes, rules, or defaults.
- Adding users who should have access to all databases on Adaptive Server.
- Granting default privileges, particularly for "guest" accounts.
- Setting database options such as select into/bulkcopy/pllsort. The settings will be reflected in all new databases. Their original value in model is off. For more information about the database options, see Chapter 22, "Setting Database Options."

Typically, most users do not have permission to modify the model database. There is not much point in granting read permission either, since Adaptive Server copies its entire contents into each new user database.

The size of model cannot be larger than the size of tempdb. Adaptive Server displays an error message if you try to increase the size of model without making tempdb at least as large.

Note Keep a backup copy of the model database, and back up model with dump database each time you change it. In case of media failure, restore model as you would a user database.

sybsystemprocs database

Sybase system procedures are stored in the database sybsystemprocs. When a user in any database executes any stored procedure, Adaptive Server first looks for that procedure in the user's current database. If there is no procedure there with that name, Adaptive Server looks for it in sybsystemprocs. If there is no procedure in sybsystemprocs by that name, Adaptive Server looks for the procedure in master.

If the procedure modifies system tables (for example, sp_adduser modifies the sysusers table), the changes are made in the database from which the procedure was executed.

To change the default permissions on system procedures, you must modify those permissions in sybsystemprocs.

Note Any time you make changes to sybsystemprocs, you should back up the database.

tempdb database

Adaptive Server has a **temporary database**, tempdb. It provides a storage area for temporary tables and other temporary working storage needs. The space in tempdb is shared among all users of all databases on the server.

The default size of tempdb depends on the logical page size for your server, either 2, 4, 8, or 16K. Certain activities may make it necessary to increase the size of tempdb. The most common of these are:

- Large temporary tables.
- A lot of activity on temporary tables, which fills up the tempdb logs.
- Large sorts or many simultaneous sorts. Subqueries and aggregates with group by also cause some activity in tempdb.

You can increase the size of tempdb with alter database. tempdb is initially created on the master device. Space can be added from the master device or from any other database device.

Creating temporary tables

No special permissions are required to use tempdb, that is, to create temporary tables or to execute commands that may require storage space in the temporary database.

Create temporary tables either by preceding the table name in a create table statement with a pound sign (#) or by specifying the name prefix “tempdb..”.

Temporary tables created with a pound sign are accessible only by the current Adaptive Server session: users on other sessions cannot access them. These nonsharable, temporary tables are destroyed at the end of each session. The first 13 bytes of the table's name, including the pound sign (#), must be unique. Adaptive Server assigns the names of such tables a 17-byte number suffix. (You can see the suffix when you query tempdb..sysobjects.)

Temporary tables created with the "tempdb.." prefix are stored in tempdb and can be shared among Adaptive Server sessions. Adaptive Server does not change the names of temporary tables created this way. The table exists either until you restart Adaptive Server or until its owner drops it using drop table.

System procedures work on temporary tables, but only if you use them from tempdb.

If a stored procedure creates temporary tables, the tables are dropped when the procedure exits. Temporary tables can also be dropped explicitly before a session ends.

Warning! Do not create temporary tables with the "tempdb.." prefix from inside a stored procedure unless you intend to share those tables among other users and sessions.

Each time you restart Adaptive Server, it copies model to tempdb, which clears the database. Temporary tables are not recoverable.

sybsecurity database

The sybsecurity database contains the audit system for Adaptive Server. It consists of:

- The system tables, sysaudits_01, sysaudits_02, ... sysaudits_08, which contain the audit trail
- The sysauditoptions table, which contains rows describing the global audit options
- All other default system tables that are derived from model

The audit system is discussed in more detail in Chapter 12, "Auditing."

sybsystemdb database

The sybsystemdb database stores information about distributed transactions. Adaptive Server versions 12.0 and later can provide transaction coordination services for transactions that are propagated to remote servers using remote procedure calls (RPCs) or Component Integration System (CIS). Information about remote servers participating in distributed transactions is stored in the syscoordinations table.

Note Adaptive Server version 12.0 and later distributed transaction management services are available as a separately-licensed feature. You must purchase and install a valid license for Distributed Transaction Management before it can be used. See *Using Adaptive Server Distributed Transaction Management Features* and the installation guide for more information.

The sybsystemdb database also stores information about SYB2PC transactions that use the Sybase two-phase commit protocol. The spt_committab table, which stores information about and tracks the completion status of each two-phase commit transaction, is stored in the sybsystemdb database.

Two-phase commit transactions and how to create the sybsystemdb database is discussed in detail in the configuration documentation for your platform.

pubs2 and *pubs3* sample databases

Installing the pubs2 and pubs3 sample databases is optional. These databases are provided as a learning tool for Adaptive Server. The pubs2 sample database is used for most of the examples in the Adaptive Server documentation, except for examples, where noted, that use the pubs3 database. For information about installing pubs2 and pubs3, see the installation documentation for your platform. For information about the contents of these sample databases, see the *Transact-SQL User's Guide*.

Maintaining the sample databases

The sample databases contain a "guest" user that allows access to the database by any authorized Adaptive Server user. The "guest" user has been given a wide range of privileges in pubs2 and pubs3, including permissions to select, insert, update, and delete user tables. For more information about the "guest" user and a list of the guest permissions in pubs2 and pubs3, see Chapter 10, "Managing Adaptive Server Logins and Database Users."

The size of the pubs2 and pubs3 databases are determined by the size of the logical page size for your server, 2, 4, 8, and 16K. If possible, you should give each new user a clean copy of pubs2 and pubs3 so that she or he is not confused by other users' changes. If you want to place pubs2 or pubs3 on a specific database device, edit the installation script before installing the database.

If space is a problem, you can instruct users to issue the begin transaction command before updating a sample database. After the user has finished updating one of the sample databases, he or she can issue the rollback transaction command to undo the changes.

pubs2 image data

Adaptive Server includes a script for installing image data in the pubs2 database (pubs3 does not use the image data). The image data consists of six pictures, two each in PICT, TIF, and Sun raster file formats. Sybase does not provide any tools for displaying image data. You must use the appropriate screen graphics tools to display the images after you extract them from the database.

See the the installation documentation for your platform for information about installing the image data in pubs2.

dbccdb database

dbcc checkstorage records configuration information for the **target database**, operation activity, and the results of the operation in the dbccdb database. Stored in the database are dbcc stored procedures for creating and maintaining dbccdb and for generating reports on the results of dbcc checkstorage operations. For more information, see Chapter 25, "Checking Database Consistency."

sybdiag database

Sybase Technical Support may create the sybdiag database on your system for debugging purposes. This database holds diagnostic configuration data, and should not be used by customers.

CHAPTER 3

System Administration for Beginners

This chapter:

- Introduces new System Administrators to important topics
- Helps System Administrators find information in the Sybase documentation

Topics include:

Experienced administrators may also find this chapter useful for organizing their ongoing maintenance activities.

Using “test” servers

It is always best to install and use a “test” and/or “development” Adaptive Server, then remove it before you create the “production” server. Using a test server makes it easier to plan and test different configurations and less stressful to recover from mistakes. It is much easier to learn how to install and administer new features when there is no risk of having to restart a production server or re-create a production database.

If you decide to use a test server, we suggest that you do so from the point of installing or upgrading Adaptive Server through the process of configuring the server. It is in these steps that you make some of the most important decisions about your final production system. The following sections describe the ways in which using a test server can help System Administrators.

Understanding new procedures and features

Using a test server allows you to practice basic administration procedures before performing them in a production environment. If you are a new Adaptive Server administrator, many of the procedures discussed in this book may be unfamiliar to you, and it may take several attempts to complete a task successfully. However, even experienced administrators will benefit from practicing techniques that are introduced by new features in Adaptive Server.

Planning resources

Working with a test server helps you plan the final resource requirements for your system and helps you discover resource deficiencies that you might not have anticipated.

In particular, disk resources can have a dramatic effect on the final design of the production system. For example, you may decide that a particular database requires nonstop recovery in the event of a media failure. This would necessitate configuring one or more additional database devices to mirror the critical database. Discovering these resource requirements in a test server allows you to change the physical layout of databases and tables without affecting database users.

You can also use a test server to benchmark both Adaptive Server and your applications using different hardware configurations. This allows you to determine the optimal setup for physical resources at both the Adaptive Server level and the operating system level before bringing the entire system online for general use.

Achieving performance goals

Most performance objectives can be met only by carefully planning a database's design and configuration. For example, you may discover that the insert and I/O performance of a particular table is a bottleneck. In this case, the best course of action may be to re-create the table on a dedicated segment and partition the table. Changes of this nature are disruptive to a production system; even changing a configuration parameter may require you to restart Adaptive Server.

Installing Sybase products

The responsibility for installing Adaptive Server and other Sybase products is sometimes placed with the System Administrator. If installation is one of your responsibilities, use the following pointers to help you in the process.

Check product compatibility

Before installing new products or upgrading existing products, always read the release bulletin included with the products to understand any compatibility issues that might affect your system. Compatibility problems can occur between hardware and software and between different release levels of the same software. Reading the release bulletin in advance can save the time and guesswork of troubleshooting known compatibility problems.

Also, refer to the lists of known problems that are installed with Adaptive Server. See the release bulletin for more information.

Install or upgrade Adaptive Server

Read through the installation documentation for your platform before you begin a new installation or upgrade. You need to plan parts of the installation and configure the operating system *before* installing Adaptive Server. It is also helpful to consult with the operating system administrator to discuss operating system requirements for Adaptive Server. These requirements can include the configuration of memory, raw devices, asynchronous I/O, and other features, depending on the platform you use. Many of these tasks must be completed before you have begun the installation.

If you are upgrading a server, back up all data (including the master database, user databases, triggers, and system procedures) offline before you begin. After upgrading, immediately create a separate, full backup of your data, especially if there are incompatibilities between older dump files and the newer versions.

Install additional third-party software

Network protocols

Adaptive Server generally includes support for the network protocol(s) that are common to your hardware platform. If your network supports additional protocols, install the required protocol support.

Directory services

As an alternative to the Sybase interfaces file, you can use a directory service to obtain a server's address and other network information. Directory services are provided by platform or third-party vendors and must be purchased and installed separately from the installation of Adaptive Server. For more information on directory services currently supported by Adaptive Server, see the configuration documentation for your platform. See also "Directory services" on page 15.

Configure and test client connections

A successful client connection depends on the coordination of Adaptive Server, the client software, and network products. If you are using one of the network protocols installed with Adaptive Server, see the configuration documentation for your platform for information about testing network connections. If you are using a different network protocol, follow the instructions that are included with the network product. You can also use "ping" utilities that are included with Sybase connectivity products to test client connections with Adaptive Server. For a general description of how clients connect to Adaptive Server, see "Connecting to Adaptive Server*" on page 14. See also the configuration documentation for your platform for details about the name and contents of the interfaces file.

Allocating physical resources

Allocating physical resources is the process of giving Adaptive Server the memory, disk space, worker processes, and CPU power required to achieve your performance and recovery goals. When installing a new server, every System Administrator must make decisions about resource utilization. You will also need to reallocate Adaptive Server's resources if you upgrade your platform by adding new memory, disk controllers, or CPUs, or if the design of your database system changes. Or, early benchmarking of Adaptive Server and your applications can help you spot deficiencies in hardware resources that create performance bottlenecks.

See Chapter 15, "Overview of Disk Resource Issues," in this manual to understand the kinds of disk resources required by Adaptive Server. See also Chapter 18, "Configuring Memory," and Chapter 20, "Managing Multiprocessor Servers," for information about memory and CPU resources.

The following sections provide helpful pointers in determining physical resource requirements.

Dedicated vs. shared servers

The first step in planning Adaptive Server resources is understanding the resources required by *other* applications running on the same machine. In most cases, System Administrators dedicate an entire machine for Adaptive Server use. This means that only the operating system and network software consume resources that otherwise might be reserved for Adaptive Server. On a shared system, other applications, such as Adaptive Server client programs or print servers, run on the same machine as Adaptive Server. It can be difficult to calculate the resources available to Adaptive Server on a shared system, because the types of programs and their pattern of use may change over time.

In either case, it is the System Administrator's responsibility to take into account the resources used by operating systems, client programs, windowing systems, and so forth when configuring resources for Adaptive Server. Configure Adaptive Server to use only the resources that are available to it. Otherwise, the server may perform poorly or fail to start.

Decision support and OLTP applications

Adaptive Server contains many features that optimize performance for OLTP, decision support, and mixed workload environments. However, you must determine in advance the requirements of your system's applications to make optimal use of these features.

For mixed workload systems, list the individual tables that you anticipate will be most heavily used for each type of application; this can help you achieve maximum performance for applications.

Advance resource planning

It is extremely important that you understand and plan resource usage in advance. In the case of disk resources, for example, after you initialize and allocate a device to Adaptive Server, that device cannot be used for any other purpose (even if Adaptive Server never fills the device with data). Likewise, Adaptive Server automatically reserves the memory for which it is configured, and this memory cannot be used by any other application.

The following suggestions can help you plan resource usage:

- For recovery purposes, it is *always* best to place a database's transaction log on a separate physical device from its data. See Chapter 21, "Creating and Managing User Databases."
- Consider mirroring devices that store mission-critical data. See Chapter 17, "Mirroring Database Devices." You may also consider using disk arrays and disk mirroring for Adaptive Server data if your operating system supports these features.
- If you are working with a test Adaptive Server, it is sometimes easier to initialize database devices as operating system files, rather than raw devices, for convenience. Adaptive Server supports either raw partitions or certified file systems for its devices.
- Keep in mind that changing configuration options can affect the way Adaptive Server consumes physical resources. This is especially true of memory resources. See Chapter 5, "Setting Configuration Parameters," for details about the amount of memory used by individual parameters.

Operating system configuration

Once you have determined the resources that are available to Adaptive Server and the resources you require, configure these physical resources at the operating system level:

- If you are using raw partitions, initialize the raw devices to the sizes required by Adaptive Server. Note that, if you initialize a raw device for Adaptive Server, that device cannot be used for any other purpose (for example, to store operating system files). Ask your operating system administrator for assistance in initializing and configuring raw devices to the required sizes.
- Configure the number of network connections. Make sure that the machine on which Adaptive Server runs can actually support the number of connections you configure. See your operating system documentation.
- Additional configuration may be required for your operating system and the applications that you use. Read the installation documentation for your platform to understand the Adaptive Server operating system requirements. Also read your client software documentation or consult with your engineers to understand the operating system requirements for your applications.

Backup and recovery

Making regular backups of your databases is crucial to the integrity of your database system. Although Adaptive Server automatically recovers from system crashes (for example, power outages) or server crashes, only *you* can recover from data loss caused by media failure. Follow the basic guidelines below for backing up your system.

The following chapters describe how to develop and implement a backup and recovery plan:

- Chapter 26, “Developing a Backup and Recovery Plan”
- Chapter 27, “Backing Up and Restoring User Databases”
- Chapter 28, “Restoring the System Databases”
- Chapter 29, “Managing Free Space with Thresholds”

Keep up-to-date backups of master

Backing up the master database is the cornerstone of any backup and recovery plan. The master database contains details about the structure of your entire database system. Its keeps track of the Adaptive Server databases, devices, and device fragments that make up those databases. Because Adaptive Server needs this information during recovery, it is crucial to maintain an up-to-date backup copy of the master database at all times.

To ensure that your backup of master is always up to date, back up the database after each command that affects disks, storage, databases, or segments. This means you should back up master after performing any of the following procedures:

- Creating or deleting databases
- Initializing new database devices
- Adding new dump devices
- Using any device mirroring command
- Creating or dropping system stored procedures, if they are stored in master
- Creating, dropping, or modifying a segment
- Adding new Adaptive Server logins

To back up master to a tape device, start isql and enter the command:

```
dump database master to "tape_device"
```

where *tape_device* is the name of the tape device (for example, */dev/rmt0*).

Keep offline copies of system tables

In addition to backing up master regularly, keep offline copies of the contents of the following system tables: sysdatabases, sysdevices, sysusages, sysloginroles, and syslogins. Do this by using the bcp utility described in the *Utility Guide*, and by storing a printed copy of the contents of each system table. You can create a printed copy by printing the output of the following queries:

```
select * from sysusages order by vstart
select * from sysdatabases
select * from sysdevices
select * from sysloginroles
select * from syslogins
```

If you have copies of these tables, and a hard disk crash or some other disaster makes your database unusable, you will be able to use the recovery procedures described in Chapter 28, "Restoring the System Databases."

You should also keep copies of all data definition language (DDL) scripts for user objects, as described under "Keeping records" on page 42.

Automate backup procedures

Creating an automated backup procedure takes the guesswork out of performing backups and makes the procedure easier and quicker to perform. Automating backups can be as simple as using an operating system script or a utility (for example, the UNIX cron utility) to perform the necessary backup commands. Or you can automate the procedure further using thresholds, which are discussed in Chapter 29, "Managing Free Space with Thresholds."

Although the commands required to create an automated script vary, depending on the operating system you use, all scripts should accomplish the same basic steps:

1 Start isql and dump the transaction log to a holding area (for example, a temporary file).

2 Rename the dump file to a name that contains the dump date, time, and database name.

3 Make a note about the new backup in a history file.

4 Record any errors that occurred during the dump in a separate error file.

5 Automatically send mail to the System Administrator for any error conditions.

Verify data consistency before backing up a database

Having backups of a database sometimes is not enough—you must have consistent, *accurate* backups (especially for master). If you back up a database that contains internal errors, the database will have the same errors when you restore it.

Using the dbcc commands, you can check a database for errors before backing it up. Always use dbcc commands to verify the integrity of a database before dumping it. If dbcc detects errors, correct them before dumping the database.

Over time, you can begin to think of running dbcc as insurance for your databases. If you discovered few or no errors while running dbcc in the past, you may decide that the risk of database corruption is small and that dbcc needs to be run only occasionally. Or, if the consequences of losing data are too high, you should continue to run dbcc commands each time you back up a database.

Note For performance considerations, many sites choose to run dbcc checks outside of peak hours or on separate servers.

See Chapter 25, "Checking Database Consistency," for information about the dbcc command.

Monitor the log size

When the transaction log becomes nearly full, it may be impossible to use standard procedures to dump transactions and reclaim space. The System Administrator should monitor the log size and perform regular transaction log dumps (in addition to regular database dumps) to make sure this situation never occurs. Use the preferred method of setting up a threshold stored procedure that notifies you (or dumps the log) when the log reaches a certain capacity. See Chapter 29, "Managing Free Space with Thresholds," for information about using threshold procedures. It is also good to dump the transaction log just prior to doing a full database dump in order to shorten the time required to dump and load the database.

You can also monitor the space used in the log segment manually by using the sp_helpsegment stored procedure, as described under "Getting information about segments" on page 698.

Ongoing maintenance and troubleshooting

In addition to making regularly scheduled backups, the System Administrator performs the following maintenance activities throughout the life of a server.

Starting and stopping Adaptive Server

Most System Administrators automate the procedure for starting Adaptive Server to coincide with the start-up of the server machine. This can be accomplished by editing operating system start-up scripts or through other operating system procedures. See the configuration documentation for your platform to determine how to start and stop Adaptive Server.

Viewing and pruning the error log

You should examine the contents of the error log on a regular basis to determine if any serious errors have occurred. You can also use operating system scripts to scan the error log for particular messages and to notify the System Administrator when specific errors occur. Checking the error log regularly helps you determine whether there are continuing problems of the same nature or whether a particular database device is going bad. See Chapter 4, "Diagnosing System Problems," for more information about error messages and their severity.

The error log file can grow large over time, since Adaptive Server appends informational and status messages to it each time it starts up. You can periodically "prune" the log file by opening the file and deleting old records. Keeping the log file to a manageable size saves disk space and makes it easier to locate current errors.

Keeping records

Keeping records about your Adaptive Server system is an important part of your job as a System Administrator. Accurate records of changes and problems that you have encountered can be a valuable reference when you are contacting Sybase Technical Support or recovering databases. More important, they can provide vital information for administrators who manage the Adaptive Server system in your absence. The following sections describe the kinds of records that are most valuable to maintain.

Contact information

Maintain a list of contact information for yourself as well as the System Security Officer, operator, and database owners on your system. Also, record secondary contacts for each role. Make this information available to all Adaptive Server users so that the appropriate contacts receive enhancement requests and problem reports.

Configuration information

Ideally, you should create databases, create database objects, and configure Adaptive Server using script files that you later store in a safe place. Storing the script files use makes it possible to re-create your entire system in the event of a disaster. It also allows you to re-create database systems quickly on new hardware platforms for evaluation purposes. If you use a third-party tool to perform system administration, remember to generate equivalent scripts after performing administration tasks.

Consider recording the following kinds of information:

- Commands used to create databases and database objects (DDL scripts)
- Commands that add new Adaptive Server logins and database users
- The current Adaptive Server configuration file, as described in "Using sp_configure with a configuration file" on page 83
- The names, locations, and sizes of all files and raw devices initialized as database devices

It is also helpful to maintain a dated log of all changes to the Adaptive Server configuration. Mark each change with a brief description of when and why you made the change, as well a summary of the end result.

Maintenance schedules

Keep a calendar of regularly scheduled maintenance activities. Such a calendar should list any of the procedures you perform at your site:

- Using dbcc to check database consistency
- Backing up user and system databases
- Monitoring the space left in transaction logs (if this is not done automatically)
- Dumping the transaction log
- Examining the error log contents for Adaptive Server, Backup Server™, and Adaptive Server Monitor™.
- Running the update statistics command (see Chapter 34, "Using the set statistics Commands," in the *Performance and Tuning Guide*)
- Examining auditing information, if the auditing option is installed

- Recompiling stored procedures
- Monitoring the resource utilization of the server machine

System information

Record information about the hardware and operating system on which you run Adaptive Server. This can include:

- Copies of operating system configuration files or start-up files
- Copies of network configuration files (for example, the *hosts* and *services* files)
- Names and permissions for the Adaptive Server executable files and database devices
- Names and locations of the tape devices used for backups
- Copies of operating system scripts or programs for automated backups, starting Adaptive Server, or performing other administration activities

Disaster recovery plan

Consolidate the basic backup and recovery procedures, the hints provided in "Backup and recovery" on page 38, and your personal experiences in recovering data into a concise list of recovery steps tailored to your system. This can be useful to both yourself and to other System Administrators who may need to recover a production system in the event of an emergency.

Getting more help

The amount of new information that System Administrators must learn may seem overwhelming. There are several software tools that can help you learn and facilitate basic administration tasks. These include Adaptive Server Monitor, used for monitoring server performance and other activities, and Sybase Central, which simplifies many administration tasks. Also available are many third-party software packages designed to help System Administrators manage daily maintenance activities.

CHAPTER 4

Diagnosing System Problems

This chapter discusses diagnosing and fixing system problems.

Topics covered in this chapter include:

How Adaptive Server uses error messages to respond to system problems

When Adaptive Server encounters a problem, it displays information—in an error message that describes whether the problem is caused by the user or the system—about the problem, how serious it is, and what you can do to fix it. The error message consists of:

- A **message number**, which uniquely identifies the error message
- A **severity level number** between 10 and 24, which indicates the type and severity of the problem
- An **error state number**, which allows unique identification of the line of Adaptive Server code at which the error was raised
- An **error message**, which tells you what the problem is, and may suggest how to fix it

For example, this is what happens if you try to access a table that does not exist:

```
select * from publisher
Msg 208, Level 16, State 1:
publisher not found. Specify owner.objectname or use
sp_help to check whether the object exists (sp_help
may produce lots of output).
```

In some cases, there can be more than one error message for a single query. If there is more than one error in a batch or query, Adaptive Server usually reports only the first one. Subsequent errors are reported the next time you execute the batch or query.

The error messages are stored in master..sysmessages, which is updated with each new release of Adaptive Server. Here are the first few rows (from an Adaptive Server with us_english as the default language):

```
select error, severity, description
from sysmessages
where error >=101 and error <=106
and langid is null
error severity description
----- -------- ----------------------------------------------------
  101       15 Line %d: SQL syntax error.
  102       15 Incorrect syntax near '%.*s'.
  103       15 The %S_MSG that starts with '%.*s' is too long.
               Maximum length is %d.
  104       15 Order-by items must appear in the select-list if
               the statement contains set operators.
  105       15 Unclosed quote before the character string '%.*s'.
  106       16 Too many table names in the query. The maximum
               allowable is %d.

(6 rows affected)
```

You can generate your own list by querying sysmessages. Here is some additional information for writing your query:

- If your server supports more than one language, sysmessages stores each message in each language. The column langid is NULL for us_english and matches the syslanguages.langid for other languages installed on the server. For information about languages on your server, use sp_helplanguage.
- The dlevel column in sysmessages is currently unused.
- The sqlstate column stores the SQLSTATE value for error conditions and exceptions defined in ANSI SQL92.

- Message numbers 17000 and greater are system procedure error messages and message strings.

Error messages and message numbers

The combination of message number (*error*) and language ID (*langid*) uniquely identifies each error message. Messages with the same message number but different language IDs are translations.

```
select error, description, langid
from sysmessages
where error = 101
error description                                langid
----- ---------------------------------------- ------
  101 Line %d: SQL syntax error.                   NULL
  101 Ligne %1!: erreur de syntaxe SQL.               1
  101 Zeile %1!: SQL Syntaxfehler.                    2

(3 rows affected)
```

The error message text is a description of the problem. The descriptions often include a line number, a reference to a kind of database object (a table, column, stored procedure, and so forth), or the name of a particular database object.

In the description field of sysmessages, a percent sign (%) followed by a character or character string serves as a placeholder for these pieces of data, which Adaptive Server supplies when it encounters the problem and generates the error message. “%d” is a placeholder for a number; “%S_MSG” is a placeholder for a kind of database object; “%.*s”—all within quotes—is a placeholder for the name of a particular database object. Table 4-1 lists placeholders and what they represent.

For example, the description field for message number 103 is:

```
The %S_MSG that starts with '%.*s' is too long.
Maximum length is %d.
```

The actual error message as displayed to a user might be:

```
The column that starts with 'title' is too long.
Maximum length is 80.
```

For errors that you report to Technical Support, it is important that you include the numbers, object types, and object names. (See “Reporting errors” on page 59.)

Variables in error message text

Table 4-1 explains the symbols that appear in the text provided with each error message explanation:

Table 4-1: Error text symbols key

Symbol	Stands For
%d, %D	Decimal number
%x,%X,%.*x,%lx, %04x, %08lx	Hexadecimal number
%s	Null-terminated string
%.*s, %*s, %*.s	String, usually the name of a particular database object
%S_*type*	Adaptive Server-defined structure
%c	Single character
%f	Floating-point number
%ld	Long decimal
%lf	Double floating-point number

Adaptive Server error logging

Error messages from Adaptive Server are sent only to the user's screen.

The backtrace from fatal error messages (severity levels 19 and higher) and error messages from the kernel are also sent to an error log file. The name of this file varies; see the configuration documentation for your platform or the *Utility Guide*.

Note The error log file is owned by the user who installed Adaptive Server (or the person who started Adaptive Server after an error log was removed). Permissions or ownership problems with the error log at the operating system level can block successful start-up of Adaptive Server.

Adaptive Server creates an error log for you if one does not already exist. You specify the location of the error log at start-up with the *errorlogfile* parameter in the runserver file or at the command line. The Sybase installation utility configures the runserver file with *$SYBASE/install* as the location of the error log if you do not choose an alternate location during installation. If you do not specify the location in the runserver file or at the command line, the location of the error log is the directory from which you start Adaptive Server. For more information about specifying the location of the error log, see description of the dataserver command in the *Utility Guide*.

Note Always start Adaptive Server from the same directory, or with the runserver file or the error log flag, so that you can locate your error log.

Each time you start a server, messages in the error log provide information on the success (or failure) of the start and the recovery of each database on the server. Subsequent fatal error messages and all kernel error messages are appended to the error log file. If you need to reduce the size of the error log by deleting old or unneeded messages, you must "prune" the log while Adaptive Server is shut down.

Error log format

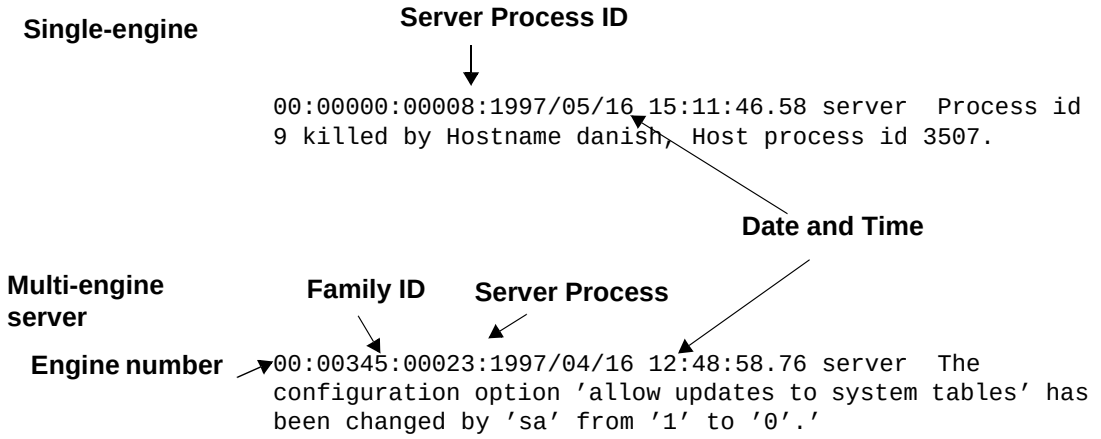
Entries in the error log include the following information:

- The engine involved for each log entry. The engine number is indicated by a 2-digit number. If only one engine is online, the display is "00."
- The family ID of the originating thread:
  - In serial processing, the display is "00000."
  - In **parallel processing**, the display is the server process ID number of the parent of the originating thread.
- The server process ID of the originating thread:
  - In serial processing, this is the server process ID number of the thread that generated the message. If the thread is a system task, then the display is "00000."
  - In parallel processing, this is the server process ID number of the originating thread.

- The date, displayed in the format *yyyy/mm/dd*, which allows you to sort error messages by date.
- The time, displayed in 24-hour format, which includes seconds and hundredths of a second.
- The word “server” or “kernel.” This entry is for Sybase Technical Support use only.
- The error message itself.

Figure 4-1 shows two examples of a line from an error log:

Figure 4-1: Error log format



Severity levels

The severity level of a message indicates information about the type and severity of the problem that Adaptive Server has encountered. For maximum integrity, when Adaptive Server responds to error conditions, it displays messages from sysmessages, but takes action according to an internal table. A few corresponding messages differ in severity levels, so you may occasionally notice a difference in expected behavior if you are developing applications or procedures that refer to Adaptive Server messages and severity levels.

Warning! You can create your own error numbers and messages based on Adaptive Server error numbers (for example, by adding 20,000 to the Adaptive Server value). However, you cannot alter the Adaptive Server-supplied system messages in the sysmessages system table.

You can add user-defined error messages to sysusermessages with sp_addmessage. See the *Adaptive Server Reference Manual*.

Users should inform the System Administrator whenever problems that generate severity levels of 17 and higher occur. The System Administrator is responsible for resolving them and tracking their frequency.

If the problem has affected an entire database, the System Administrator may have to use the database consistency checker (dbcc) to determine the extent of the damage. The dbcc may identify some objects that have to be removed. It can repair some damage, but the database may have to be reloaded.

For more information, refer to the following chapters:

- dbcc is discussed in Chapter 25, "Checking Database Consistency."
- Loading a user database is discussed in Chapter 27, "Backing Up and Restoring User Databases."
- Loading system databases is discussed in Chapter 28, "Restoring the System Databases."

The following sections discuss each severity level.

Levels 10–18

Error messages with severity levels 10–16 are generated by problems that are caused by user errors. These problems can always be corrected by the user. Severity levels 17 and 18 do not terminate the user's session.

Error messages with severity levels 17 and higher should be reported to the System Administrator or Database Owner.

Level 10: Status information

Messages with severity level 10 are not errors at all. They provide additional information after certain commands have been executed and, typically, do not display the message number or severity level. For example, after a create database command has been run, Adaptive Server displays a message telling the user how much of the requested space has been allocated for the new database.

Level 11: Specified database object not found

Messages with severity level 11 indicate that Adaptive Server cannot find an object that was referenced in the command.

This is often because the user has made a mistake in typing the name of a database object, because the user did not specify the object owner's name, or because of confusion about which database is current. Check the spelling of the object names, use the owner names if the object is not owned by you or "dbo," and make sure you are in the correct database.

Level 12: Wrong datatype encountered

Messages with severity level 12 indicate a problem with datatypes. For example, the user may have tried to enter a value of the wrong datatype in a column or to compare columns of different and incompatible datatypes.

To correct comparison problems, use the convert function with select. For information on convert, see the *Adaptive Server Reference Manual* or the *Transact-SQL User's Guide*.

Level 13: User transaction syntax error

Messages with severity level 13 indicate that something is wrong with the current user-defined transaction. For example, you may have issued a commit transaction command without having issued a begin transaction or you may have tried to roll back a transaction to a savepoint that has not been defined (sometimes there may be a typing or spelling mistake in the name of the savepoint).

Severity level 13 can also indicate a deadlock, in which case the deadlock victim's process is rolled back. The user must restart his or her command.

Level 14: Insufficient permission to execute command

Messages with severity level 14 mean that you do not have the necessary permission to execute the command or access the database object. You can ask the owner of the database object, the owner of the database, or the System Administrator to grant you permission to use the command or object in question.

Level 15: Syntax error in SQL statement

Messages with severity level 15 indicate that the user has made a mistake in the syntax of the command. The text of these error messages includes the line numbers on which the mistake occurs and the specific word near which it occurs.

Level 16: Miscellaneous user error

Most error messages with severity level 16 reflect that the user has made a nonfatal mistake that does not fall into any of the other categories. Severity level 16 and higher can also indicate software or hardware errors.

For example, the user may have tried to update a view in a way that violates the restrictions. Another error that falls into this category is unqualified column names in a command that includes more than one table with that column name. Adaptive Server has no way to determine which one the user intends. Check the command syntax and working database context.

Messages that ordinarily have severities greater than 16 will show severity 16 when they are raised by dbcc checktable or dbcc checkalloc so that checks can continue to the next object. When you are running the dbcc utility, check the *Error Messages* manual for information about error messages between 2500 and 2599 with a severity level of 16.

Note Levels 17 and 18 are usually not reported in the error log. Users should be instructed to notify the System Administrator when level 17 and 18 errors occur.

Level 17: Insufficient resources

Error messages with severity level 17 mean that the command has caused Adaptive Server to run out of resources or to exceed some limit set by the System Administrator. You can continue with the work you are doing, although you may not be able to execute a particular command.

These system limits include the number of databases that can be open at the same time and the number of connections allowed to Adaptive Server. They are stored in system tables and can be checked with sp_configure. See Chapter 5, "Setting Configuration Parameters," for more information on changing configuration variables.

The Database Owner can correct the level 17 error messages indicating that you have run out of space. Other level 17 error messages should be corrected by the System Administrator.

Level 18: Non-fatal internal error detected

Error messages with severity level 18 indicate some kind of internal software bug. However, the command runs to completion, and the connection to Adaptive Server is maintained. You can continue with the work you are doing, although you may not be able to execute a particular command. An example of a situation that generates severity level 18 is Adaptive Server detecting that a decision about the access path for a particular query has been made without a valid reason.

Since problems that generate such messages do not keep users from their work, users tend not to report them. Users should be instructed to inform the System Administrator every time an error message with this severity level (or higher) occurs so that the System Administrator can report them.

Severity levels 19–26

Fatal problems generate error messages with severity levels 19 and higher. They break the user's connection to Adaptive Server (some of the higher severity levels shut down Adaptive Server). To continue working, the user must restart the client program.

When a fatal error occurs, the process freezes its state before it stops, recording information about what was happening. It is then killed and disappears.

When the user's connection is broken, he or she may or may not be able to reconnect and resume working. Some problems with severity levels in this range affect only one user and one process. Others affect all the processes in the database. In some cases, it will be necessary to restart Adaptive Server. These problems do not necessarily damage a database or its objects, but they can. They may also result from earlier damage to a database or its objects. Other problems are caused by hardware malfunctions.

A backtrace of fatal error messages from the kernel is directed to the error log file, where the System Administrator can review it.

Level 19: Adaptive Server fatal error in resource

Error messages with severity level 19 indicate that some non-configurable internal limit has been exceeded and that Adaptive Server cannot recover gracefully. You must reconnect to Adaptive Server.

Level 20: Adaptive Server fatal error in current process

Error messages with severity level 20 indicate that Adaptive Server has encountered a bug in a command. The problem has affected only the current process, and it is unlikely that the database itself has been damaged. Run dbcc diagnostics. You must reconnect to Adaptive Server.

Level 21: Adaptive Server fatal error in database processes

Error messages with severity level 21 indicate that Adaptive Server has encountered a bug that affects all the processes in the current database. However, it is unlikely that the database itself has been damaged. Restart Adaptive Server and run the dbcc diagnostics. You must reconnect to Adaptive Server.

Level 22: Adaptive Server fatal error: Table integrity suspect

Error messages with severity level 22 indicate that the table or index specified in the message was previously damaged by a software or hardware problem.

The first step is to restart Adaptive Server and run dbcc to determine whether other objects in the database are also damaged. Whatever the report from dbcc may be, it is possible that the problem is in the cache only and not on the disk itself. If so, restarting Adaptive Server will fix the problem.

If restarting does not help, then the problem is on the disk as well. Sometimes, the problem can be solved by dropping the object specified in the error message. For example, if the message tells you that Adaptive Server has found a row with length 0 in a nonclustered index, the table owner can drop the index and re-create it.

Adaptive Server takes any pages or indexes offline that it finds to be suspect during recovery. Use sp_setsuspect_granularity to determine whether recovery marks an entire database or only individual pages as suspect. See sp_setsuspect_granularity in the Adaptive Server Reference Manual for more information.

You must reconnect to Adaptive Server.

Level 24: Hardware error or system table corruption

Error messages with severity level 24 reflect some kind of media failure or (in rare cases) the corruption of sysusages. The System Administrator may have to reload the database. It may be necessary to call your hardware vendor.

Level 23: Fatal error: Database integrity suspect

Error messages with severity level 23 indicate that the integrity of the entire database is suspect due to previous damage caused by a software or hardware problem. Restart Adaptive Server and run dbcc diagnostics.

Even when a level 23 error indicates that the entire database is suspect, the damage may be confined to the cache, and the disk itself may be fine. If so, restarting Adaptive Server with startserver will fix the problem.

Level 25: Adaptive Server internal error

Level 25 errors are not displayed to the user; this level is only used for Adaptive Server internal errors.

Level 26: Rule error

Error messages with severity level 26 reflect that an internal locking or synchronization rule was broken. You must shut down and restart Adaptive Server.

Reporting errors

When you report an error, include:

- The message number, level number, and state number.
- Any numbers, database object types, or database object names that are included in the error message.
- The context in which the message was generated, that is, which command was running at the time. You can help by providing a hard copy of the backtrace from the error log.

Backup Server error logging

Like Adaptive Server, Backup Server creates an error log if one does not already exist.You specify the location of the error log at start-up with the *error_log_file* parameter in the runserver file or at the command line. The Sybase installation utility configures the runserver file with *$SYBASE/install* as the location of the error log if +you do not choose an alternate location during installation. If you do not specify the location in the runserver file or at the command line, the location of the error log is the directory from which you start Backup Server. Use the backupserver -V option (bcksvr -V on Windows NT) to limit the messages printed to the error log. For more information about specifying the location of the error log, see the sections describing Backup Server in the *Utility Guide*.

Backup Server error messages are in the form:

```
MMM DD YYY: Backup Server:N.N.N.N: Message Text
```

Backup Server message numbers consist of 4 integers separated by periods, in the form N.N.N.N. Messages in the form N.N.N are sent by Open Server™.

The four components of a Backup Server error message are *major.minor.severity.state* :

- The *major* component generally indicates the functional area of the Backup Server code where the error occurred:
  - 1 – System errors
  - 2 – Open Server event errors
  - 3 – Backup Server remote procedure call errors
  - 4 – I/O service layer errors
  - 5 – Network data transfer errors
  - 6 – Volume handling errors
  - 7 – Option parsing errors

  Major error categories 1–6 may result from Backup Server internal errors or a variety of system problems. Major errors in category 7 are almost always due to problems in the options you specified in your dump or load command.
- *minor* numbers are assigned in order within a major category.
- *severity* is:
  - 1 – Informational, no user action necessary.
  - 2, 3 – An unexpected condition, possibly fatal to the session, has occurred. The error may have occurred with usage, environment, or internal logic, or any combination of these factors.
  - 4 – An unexpected condition, fatal to the execution of the Backup Server, has occurred. The Backup Server must exit immediately.
- *state* codes have a one-to-one mapping to instances of the error report within the code. If you need to contact Technical Support about Backup Server errors, the state code helps determine the exact cause of the error.

Killing processes

A process is a unit of execution carried out by Adaptive Server. Each process is assigned a unique process identification number when it starts, this number is called a spid. These numbers are stored, along with other information about each process, in master..sysprocesses. Processes running in a parallel processes environment create child processes, each of which has its own spids. Several processes create and assign spids: booting Adaptive Server, login tasks, checkpoints, the housekeeper task, and so on. You can see most of the information by running sp_who.

Running sp_who on a single-engine server shows the sp_who process running and all other processes that are "runnable" or in one of the sleep states. In multi-engine servers, there can be a process running for each engine.

The kill command gets rid of an ongoing process. The most frequent reason for killing a process is that it interferes with other users and the person responsible for running it is not available. The process may hold locks that block access to database objects, or there may be many sleeping processes occupying the available user connections. A System Administrator can kill processes that are:

- Waiting for an alarm, such as a waitfor command
- Waiting for network sends or receives
- Waiting for a lock
- Waiting for synchronization messages from another process in a family
- Most running or "runnable" processes

Adaptive Server allows you to kill processes only if it can cleanly roll back any uncompleted transactions and release all system resources that are used by the process. For processes that are part of a family, killing any of the child processes will also kill all other processes in the family. However, it is easiest to kill the parent process. For a family of processes, the kill command is detected more quickly if the status of the child processes is

```
sync sleep
```

Table 4-2 shows the values that sp_who reports and when the kill command takes effect.

Table 4-2: Status values reported by sp_who

Status	Indicates	Effects of kill command
recv sleep	Waiting on a network read	Immediate.
send sleep	Waiting on a network send	Immediate.
alarm sleep	Waiting on an alarm such as `waitfor delay "10:00"`	Immediate.
lock sleep	Waiting on a lock acquisition	Immediate.
sync sleep	Waiting on a synchronization message from another process in the family.	Immediate. Other processes in the family must also be brought to state in which they can be killed
sleeping	Waiting on a disk I/O, or some other resource. Probably indicates a process that is running, but doing extensive disk I/O	Killed when it "wakes up," usually immediate; a few sleeping processes do not wake up and require a Server reboot to clear.
runnable	In the queue of runnable processes	Immediate.
running	Actively running on one of the server engines	Immediate.
infected	Server has detected serious error condition; extremely rare	kill command not recommended. Server reboot probably required to clear process.
background	A process, such as a threshold procedure, run by Adaptive Server rather than by a user process	Immediate; use kill with extreme care. Recommend a careful check of sysprocesses before killing a background process.
log suspend	Processes suspended by reaching the last-chance threshold on the log	Immediate.

Only a System Administrator can issue the kill command; permission to use it cannot be transferred.

The syntax is:

```
kill spid
```

You can kill only one process at a time, but you can perform a series of kill commands in a batch. For example:

```
1> kill 7
2> kill 8
3> kill 9
4> go
```

A kill command is not reversible and cannot be included in a user-defined transaction. spid must be a numeric constant; you cannot use a variable. Here is some sample output from sp_who:

```
fid    spid       status        loginame          origname       hostname
blk         dbname
    cmd
----     -----        --------- --------                    ---------        -
------         ----         -------            -    ----------------
0    1        recv sleep       howard          howard        svr30eng    0
master
    AWAITING COMMAND
0    2        sleeping       NULL          NULL              0        master
    NETWORK HANDLER
0    3       sleeping       NULL          NULL              0       master
    DEADLOCK TUNE
0    4       sleeping       NULL          NULL              0        master
    MIRROR HANDLER
0    5        sleeping       NULL          NULL               0        master
    CHECKPOINT SLEEP
0    6       sleeping       NULL          NULL              0        master
    HOUSEKEEPER
0   7      recv sleep      bill         bill      bigblue      0      master
    AWAITING COMMAND
0   8   recv sleep        wilbur         wilbur      hazel      0      master
    AWAITING COMMAND
0   9   recv sleep         joan         joan      luv2work      0      master
    AWAITING COMMAND
0   10   running         foote          foote       svr47hum    0      master
    SELECT
(10 rows affected, return status = 0)
```

In the example above, processes 2–6 cannot be killed: they are system processes. The login name NULL and the lack of a host name identify them as system processes. You will always see NETWORK HANDLER, MIRROR HANDLER, HOUSEKEEPER, and CHECKPOINT SLEEP (or, rarely, CHECKPOINT). AUDIT PROCESS becomes activated if you enable auditing.

Processes 1, 8, 9, and 10 can be killed, since they have the status values "recv sleep," "send sleep," "alarm sleep," and "lock sleep."

In sp_who output, you cannot tell whether a process whose status is "recv sleep" belongs to a user who is using Adaptive Server and may be pausing to examine the results of a command or whether the process indicates that a user has restarted a PC or other terminal, and left a stranded process.You can learn more about a questionable process by querying the sysprocesses table for information. For example, this query shows the host process ID and client software used by process 8:

```
select hostprocess, program_name
```

```
    from sysprocesses
where spid = 8
 hostprocess program_name
 ----------- ----------------
 3993        isql
```

This query, plus the information about the user and host from the sp_who results, provides additional information for tracking down the process from the operating system level.

Using *sp_lock* to examine blocking processes

In addition to sp_who, sp_lock can help identify processes that are blocking other processes. If the blk column in the sp_who report indicates that another process has been blocked while waiting to acquire locks, sp_lock can display information about the blocking process. For example, process 10 in the sp_who output above is blocked by process 7. To see information about process 7, execute:

```
sp_lock 7
```

For more information about locking in Adaptive Server, see the *Performance and Tuning Guide*.

Configuring Adaptive Server to save SQL batch text

Occasionally a query or procedure causes Adaptive Server Monitor to hang. Users with the System Administrator role can configure Adaptive Server to give Adaptive Server Monitor access to the text of the currently executing SQL batch. Viewing the SQL text of long-running batches helps you debug hung processes or fine-tune long statements that are heavy resource consumers.

Adaptive Server must be configured to collect the SQL batch text and write it to shared memory, where the text can be read by Adaptive Server Monitor Server (the server component of Adaptive Server Monitor). The client requests might come from Monitor Viewer, which is a plug-in to Sybase Central, or other Adaptive Server Monitor Server applications.

Configuring Adaptive Server to save SQL batch text also allows you to view the current query plan in showplan format (as you would see after setting showplan on). You can view the current query plan from within Adaptive Server; see "Viewing the query plan of a SQL statement" on page 68. SQL batches are viewable only through Adaptive Server Monitor Server. See the Adaptive Server Monitor Server documentation for more information about displaying the batch text.

Because the query or procedure you are viewing may be nested within a batch of SQL text, the sysprocesses table now includes columns for the line number, statement number and spid of a hung statement to view its query plan.

By default, Adaptive Server is not configured to save SQL batch text, so you must configure Adaptive Server to allocate memory for this feature. Adaptive Server Monitor access to SQL has no effect on performance if you have not configured any memory to save SQL batches.

Allocating memory for batch text

You can configure the amount of the SQL text batch you want to save. When text saving is enabled, Adaptive Server copies the subsequent SQL text batches to memory shared with SQL Server Monitor. Because each new batch clears the memory for the connection and overwrites the previous batch, you can view only currently executing SQL statements. To save SQL text:

1 Configure the amount of SQL text retained in memory (see "Configuring the amount of SQL text retained in memory" on page 65).

2 Enable SQL Server to start saving SQL text (see "Enabling Adaptive Server to start saving SQL text" on page 67).

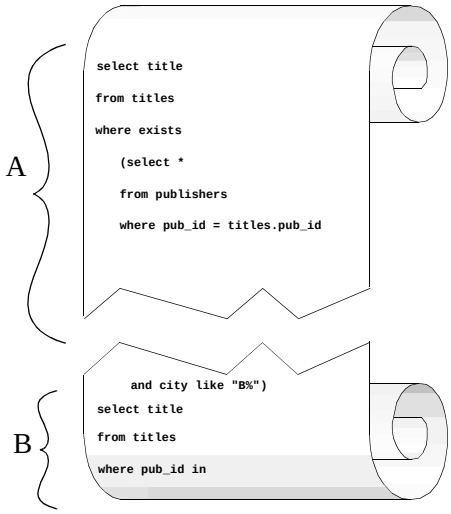
Note You must have System Administration privileges to configure and save SQL text batches.

Configuring the amount of SQL text retained in memory

After installation, you must decide the maximum amount of SQL text that can be copied to shared memory. Consider the following to help you determine how much memory to allocate per user:

- SQL batches exceeding the allocated amount of memory are truncated without warning. If you do not allocate enough memory for the batch statements, the text you are interested in viewing might be the section of the batch that is truncated, as illustrated in Figure 4-2.

Figure 4-2: How SQL text is truncated if not enough memory is configured



For example, if you configure Adaptive Server to save the amount of text designated by bracket A in the illustration, but the statement that is running occurs in the text designated by bracket B, Adaptive Server will not display the statement that is running.

- The more memory you allocate for SQL text from shared memory, the less chance the problem statement will be truncated from the batch copied to shared memory. However, Adaptive Server immediately rejects very large values because they do not leave enough memory for data and procedure caches.

Sybase recommends an initial value of 1024 bytes per user connection.

Use sp_configure with the max SQL text monitored configuration parameter to allocate shared memory:

```
sp_configure "max SQL text monitored", bytes_per_connection
```

where *bytes_per_connection* (the maximum number of bytes saved for each client connection) is between 0 (the default) and 2,147,483,647 (the theoretical limit).

Since memory for SQL text is allocated by Adaptive Server at start-up, you must restart Adaptive Server for this parameter to take effect.

The total memory allocated for the SQL text from shared memory is the product of *bytes_per_connection* multiplied by the number of user connections.

Enabling Adaptive Server to start saving SQL text

After you allocate shared memory for SQL text, Adaptive Server saves a copy of each SQL batch whenever you enable an Adaptive Server Monitor event summary that includes SQL batches.

You may also have to reconfigure Adaptive Server Monitor's event buffer scan interval for SQL text. See the Adaptive Server Monitor documentation for more information.

SQL commands not represented by text

If you use Client-Library™ functions not represented by text (such as ct_cursor or ct_dynamic) to issue SQL commands, Client-Library encodes the information for efficiency, and Adaptive Server generally decodes and displays key command information. For example, if you open a cursor with ct_cursor and the command is running, the Adaptive Server Monitor event summary displays the cursor name and the cursor declare statement.

Table 4-3 lists a complete list of the Client-Library functions not represented by text:

Table 4-3: SQL commands not represented by text

Client-Library routine	DB-Library routine	Presentation name	Presentation data
ct_cursor	N/A	CLOSE_CURSOR	Cursor name, statement
ct_cursor	N/A	DECLARE_CURSOR	Cursor name, statement
ct_cursor	N/A	DELETE_AT_CURSOR	Cursor name, statement

Client-Library routine	DB-Library routine	Presentation name	Presentation data
ct_cursor	N/A	FETCH_CURSOR	Cursor name, statement
ct_fetch (when processing the results of ct_cursor)	N/A	FETCH_CURSOR	Cursor name, statement
ct_cursor CURSOR_ROWS, or ct_cancel when the connection has Client-Library cursors	N/A	CURSOR_INFO	Cursor name, statement
ct_cursor	N/A	OPEN_CURSOR	Cursor name, statement
ct_cursor	N/A	UPDATE_AT_CURSOR	Cursor name, statement
ct_command(CS_RPC_CMD) (default behavior)	dbrpcinit (only in version 10.0.1 or later)	DBLIB_RPC	RPC name
ct_dynamic	N/A	DYNAMIC_SQL	Dynamic statement name, statement
ct_command(CS_MSG_CMD	N/A	MESSAGE	None
ct_param	dbrpcparam	PARAM_FORMAT	None
ct_param	dbrpcparam	PARAMS	None
ct_command(CS_RPC_CMD) (only when a TDS version earlier than 5.0 is used)	dbrpcparam (in DB-Library version earlier than 10.0.1)	RPC	RPC name

For more information about SQL commands not represented by text, see your Open Client documentation.

Viewing the query plan of a SQL statement

Use sp_showplan and the *spid* of the user connection in question to retrieve the query plan for the statement currently running on this connection. You can also use sp_showplan to view the query plan for a previous statement in the same batch.

Here is the syntax:

```
declare @batch int
declare @context int
declare @statement int
execute sp_showplan <spid_value>, @batch_id= @batch output,
@context_id= @context output, @stmt_num=@statement output
```

where *batch_id* is the unique number for a batch, *context_id* is a unique number for every procedure (or trigger) executed in the batch, and *stmt_num* is the number of the current statement within a batch. Adaptive Server uses the unique batch ID to synchronize the query plan with the batch text and other data retrieved by Adaptive Server Monitor.

Note You must be a System Administrator to execute sp_showplan.

For example, to see the query plan for the current statement for spid 99:

```
declare @batch int
declare @context int
declare @statement int
exec sp_showplan 99, @batch output, @context output, @statement output
```

You can run the query plan procedure independently of Adaptive Server Monitor, regardless of whether or not Adaptive Server has allocated shared memory for SQL text.

Viewing previous statements

To see the query plan for the previous statement in the same batch, issue sp_showplan with the same values as the original query, but subtract one from the statement number. Using this method, you can view all the statements in the statement batch back to query number one.

Viewing a nested procedure

Although sp_showplan allows you to view the query plan for the current statement, the actual statement that is running may exist within a procedure (or within a nested chain of procedures) called from the original SQL batch. Table 4-4 shows the columns in sysprocesses that contain information about these nested statements.

Table 4-4: Columns added to sysprocesses

Column	Datatype	Specifies
id	Integer	The object ID of the running procedure (or 0 if no procedure is running)
stmtnum	Integer	The current statement number within the running procedure (or the SQL batch statement number if no procedure is running)
linenum	Integer	The line number of the current statement within the running stored procedure (or the line number of the current SQL batch statement if no procedure is running)

This information is saved in sysprocesses, regardless of whether SQL text is enabled or any memory is allocated for SQL text.

To display the id, stmtnum, and linenum columns, enter:

```
select id, stmtnum, linenum
from sysprocesses
where spid = spid_of_hung_session
```

Note You do not need the sa_role to run this select statement.

Shutting down servers

A System Administrator can shut down Adaptive Server or Backup Server with the shutdown command. The syntax is:

shutdown [*backup_server_name*] [with {wait|nowait}]

The default for the shutdown command is with wait. That is, shutdown and shutdown with wait do exactly the same thing.

Shutting down Adaptive Server

If you do not give a server name, shutdown shuts down the Adaptive Server you are using. When you issue a shutdown command, Adaptive Server:

1 Disables logins, except for System Administrators

2 Performs a checkpoint in each database, flushing pages that have changed from memory to disk

3 Waits for currently executing SQL statements or procedures to finish

In this way, shutdown minimizes the amount of work that automatic recovery must do when you restart Adaptive Server.

The with nowait option shuts down Adaptive Server immediately. User processes are aborted, and recovery may take longer after a shutdown with nowait. You can help minimize recovery time by issuing a checkpoint command before you issue a shutdown with nowait command.

Shutting down a Backup Server

To shut down a Backup Server, give the Backup Server's name:

```
shutdown SYB_BACKUP
```

The default is with wait, so any dumps or loads in progress will complete before the Backup Server process halts. After you issue a shutdown command, no new dump or load sessions can be started on the Backup Server.

To see the names of the Backup Servers that are accessible from your Adaptive Server, execute sp_helpserver. Use the value in the name column in the shutdown command. You can shut down a Backup Server only if it is:

- Listed in sysservers on your Adaptive Server, and
- Listed in your local interfaces file.

Use sp_addserver to add a Backup Server to sysservers.

Checking for active dumps and loads

To see the activity on your Backup Server before executing a shutdown command, run sp_who on the Backup Server:

```
                SYB_BACKUP...sp_who
spid   status   loginame hostname   blk cmd
------ -------- -------- ---------- --- ----------------
     1 sleeping NULL     NULL       0   CONNECT HANDLER
     2 sleeping NULL     NULL       0   DEFERRED HANDLER
     3 runnable NULL     NULL       0   SCHEDULER
     4 runnable NULL     NULL       0   SITE HANDLER
```

```
 5 running  sa       heliotrope 0   NULL
```

Using *nowait* on a Backup Server

The shutdown *backup_server* with nowait command shuts down the Backup Server, regardless of current activity. Use it only in severe circumstances. It can leave your dumps or loads in incomplete or inconsistent states.

If you use shutdown with nowait during a log or database dump, check for the message indicating that the dump completed. If you did not receive this message, or if you are not sure whether the dump completed, your next dump should be a dump database, not a transaction dump. This guarantees that you will not be relying on possibly inconsistent dumps.

If you use shutdown with nowait during a load of any kind, and you did not receive the message indicating that the load completed, you may not be able to issue further load transaction commands on the database. Be sure to run a full database consistency check (dbcc) on the database before you use it. You may have to reissue the full set of load commands, starting with load database.

Learning about known problems

The release bulletin is a valuable resource for learning about known problems or incompatibilities with Adaptive Server and Backup Server. Reading the release bulletin in advance can save you the time and guesswork of troubleshooting known problems.

The Adaptive Server installation program also installs files that list all system problem reports (SPRs) and closed problem reports (CPRs) for Adaptive Server. Problem reports are organized by functional areas of the product. For example, a file named *cpr_bus* would contain a listing of closed (fixed) problem reports pertaining to the Backup Server, and the file *spr_bus* would contain a list of currently open problem reports for the Backup Server.

See the release bulletin to learn the location of CPR and SPR files.

CHAPTER 5

Setting Configuration Parameters

This chapter describes the Adaptive Server configuration parameters. A configuration parameter is a user-definable setting that you set with the system procedure sp_configure. Configuration parameters are used for a wide range of services, from basic to specific server operations, and for performance tuning.

Adaptive Server configuration parameters

The following table lists the Adaptive Server configuration parameters alphabetically.

Configuration parameters
“abstract plan cache” on page 176
“abstract plan dump” on page 176
“abstract plan load” on page 177
“abstract plan replace” on page 177
“additional network memory” on page 169
“allow backward scans” on page 177
“allow nested triggers” on page 178
“allow procedure grouping” on page 213
“allow remote access” on page 149
“allow resource limits” on page 178
“allow sendmsg” on page 150
“allow sql server async i/o” on page 109
“allow resource limits” on page 178
“allow updates to system tables” on page 179
“auditing” on page 213
“audit queue size” on page 213
“cis bulk insert array size” on page 105

Configuration parameters
“cis bulk insert batch size” on page 105
“cis connect timeout” on page 106
“cis cursor rows” on page 106
“cis packet size” on page 106
“cis rpc handling” on page 107
“configuration file” on page 128
“cpu accounting flush interval” on page 180
“cpu grace time” on page 181
“current audit table” on page 214
“deadlock checking period” on page 135
“deadlock retries” on page 136
“default character set id” on page 131
“default database size” on page 182
“default exp_row_size percent” on page 183
“default fill factor percent” on page 183
“default language id” on page 132
“default network packet size” on page 150
“default sortorder id” on page 132
“default unicode sortorder” on page 221
“disable character set conversions” on page 132
“disk i/o structures” on page 110
“dtm detach timeout period” on page 113
“dtm lock timeout period” on page 114
“dump on conditions” on page 184
“enable full-text search” on page 108
“enable cis” on page 108
“enable DTM” on page 115
“enable housekeeper GC” on page 190
“enable HA” on page 189
“enable java” on page 128
“enable enterprise java beans” on page 129
“enable file access” on page 108
“enable full-text search” on page 108
“enable rep agent threads” on page 175
“enable ssl” on page 215
“enable sort-merge joins and JTC” on page 185
“enable surrogate processing” on page 221

Configuration parameters
“enable unicode conversion” on page 221
“enable unicode normalization” on page 222
“enable xact coordination” on page 116
“esp execution priority” on page 124
“esp execution stacksize” on page 125
“esp unload dll” on page 125
“event buffers per engine” on page 186
“event log computer name (Windows NT only)” on page 122
“event logging (Windows NT only)” on page 123
“executable codesize + overhead” on page 141
“global async prefetch limit” on page 99
“global cache partition number” on page 100
“housekeeper free write percent” on page 187
“i/o accounting flush interval” on page 192
“i/o polling process count” on page 193
“identity burning set factor” on page 190
“identity grab size” on page 191
“license information” on page 212
“lock address spinlock ratio” on page 133
“lock hashtable size” on page 138
“lock shared memory” on page 171
“lock scheme” on page 139
“lock spinlock ratio” on page 137
“lock table spinlock ratio” on page 140
“lock wait period” on page 139
“log audit logon failure” on page 123
“log audit logon success” on page 124
“max async i/os per engine” on page 159
“max async i/os per server” on page 159
“max cis remote connections” on page 109
“max network packet size” on page 152
“max number network listeners” on page 155
“max online engines” on page 174
“max parallel degree” on page 164
“max scan parallel degree” on page 165
“max SQL text monitored” on page 171
“maximum dump conditions” on page 197

Configuration parameters
“memory alignment boundary” on page 100
“memory per worker process” on page 166
“msg confidentiality reqd” on page 216
“msg integrity reqd” on page 216
“number of alarms” on page 197
“number of aux scan descriptors” on page 198
“number of devices” on page 111
“number of dtx participants” on page 117
“number of index trips” on page 101
“number of large i/o buffers” on page 95
“number of locks” on page 134
“number of mailboxes” on page 200
“number of messages” on page 201
“number of oam trips” on page 102
“number of open databases” on page 142
“number of open indexes” on page 144
“number of open objects” on page 145
“number of pre-allocated extents” on page 201
“number of remote connections” on page 155
“number of remote logins” on page 156
“number of remote sites” on page 156
“number of sort buffers” on page 202
“number of user connections” on page 223
“number of worker processes” on page 163
“open index hash spinlock ratio” on page 147
“open index spinlock ratio” on page 148
“open object spinlock ratio” on page 148
“o/s file descriptors” on page 161
“page lock promotion HWM” on page 194
“page lock promotion LWM” on page 195
“page lock promotion PCT” on page 196
“page utilization percent” on page 112
“partition groups” on page 203
“partition spinlock ratio” on page 203
“permission cache entries” on page 225
“print deadlock information” on page 204
“print recovery information” on page 96

Configuration parameters
“procedure cache size” on page 103
“read committed with lock” on page 140
“recovery interval in minutes” on page 96
“remote server pre-read packets” on page 157
“row lock promotion HWM” on page 210
“row lock promotion LWM” on page 210
“row lock promotion PCT” on page 211
“runnable process search count” on page 205
“secure default login” on page 216
“select on syscomments.text column” on page 217
“shared memory starting address” on page 161
“size of auto identity column” on page 205
“size of global fixed heap” on page 130
“size of process object heap” on page 130
“size of shared class heap” on page 130
“size of unilib cache” on page 223
“SQL Perfmon Integration (Windows NT only)” on page 206
“sql server clock tick length” on page 207
“stack guard size” on page 226
“stack size” on page 229
“start mail session (Windows NT only)” on page 126
“strict dtm enforcement” on page 118
“suspend audit when device full” on page 218
“syb_sendmsg port number” on page 157
“systemwide password expiration” on page 219
“lock table spinlock ratio” on page 140
“tape retention in days” on page 99
“tcp no delay” on page 158
“text prefetch size” on page 208
“time slice” on page 208
“total data cache size” on page 104
“total logical memory” on page 173
“txn to pss ratio” on page 119
“unified login required (Windows NT only)” on page 220
“upgrade version” on page 209
“user log cache size” on page 230
“user log cache spinlock ratio” on page 231

Configuration parameters
“use security services (Windows NT only)” on page 220
“xact coordination interval” on page 120
“xp_cmdshell context” on page 126

What are configuration parameters?

Configuration parameters are user-definable settings that control various aspects of Adaptive Server’s behavior. Adaptive Server supplies default values for all configuration parameters. You can use configuration parameters to tailor Adaptive Server for an installation’s particular needs.

Read this chapter carefully to determine which configuration parameters you should reset to optimize server performance. Also, see the *Performance and Tuning Guide* for further information on using sp_configure to tune Adaptive Server.

Warning! Change configuration parameters with caution. Arbitrary changes in parameter values can adversely affect Adaptive Server performance and other aspects of server operation.

The Adaptive Server configuration file

Adaptive Server stores the values of configuration parameters in a configuration file, which is an ASCII text file. When you install a new Adaptive Server, your parameters are set to the default configuration; the default name of the file is *server_name.cfg*, and the default location of the file is the Sybase installation directory ($SYBASE). When you change a configuration parameter, Adaptive Server saves a copy of the old configuration file as *server_name.001*, *server_name.002*, and so on. Adaptive Server writes the new values to the file *server_name.cfg* or to a file name you specify at start-up.

How to modify configuration parameters

You set or change configuration parameters in one of the following ways:

- By executing the system procedure sp_configure with the appropriate parameters and values,
- By editing your configuration file and then invoking sp_configure with the configuration file option, or
- By specifying the name of a configuration file at start-up.

Configuration parameters are either *dynamic* or *static*. Dynamic parameters go into effect as soon as you execute sp_configure. Static parameters require Adaptive Server to reallocate memory, so they take effect only after Adaptive Server has been restarted. The description of each parameter indicates whether it is static or dynamic. Adaptive Server writes the new value to the system table sysconfigures and to the configuration file when you change the value, not when you restart Adaptive Server. The current configuration file and sysconfigures reflect configured values, not run values. The system table syscurconfigs reflects current run values of configuration parameters.

Who can modify configuration parameters

The roles required for using sp_configure are as follows:

- Any user can execute sp_configure to display information about parameters and their current values.
- Only a System Administrator and System Security Officer can execute sp_configure to modify configuration parameters.
- Only a System Security Officer can execute sp_configure to modify values for:

• allow procedure grouping	• secure default login
• allow updates to system tables	• select on syscomments.text column
• auditing	• suspend audit when device full
• audit queue size	• systemwide password expiration
• current audit table	• unified login required (Windows NT only)
• msg confidentiality reqd	• use security services (Windows NT only)
• msg confidentiality reqd	• secure default login
• msg integrity reqd	• select on syscomments.text column
• allow procedure grouping	• suspend audit when device full
• allow updates to system tables	• systemwide password expiration

• auditing	• unified login required (Windows NT only)
• audit queue size	• use security services (Windows NT only)
• current audit table	• to enable SSL session-based security
• msg confidentiality reqd	
• allow remote access	
• allow remote access	

Unit specification using sp_configure

sp_configure allows you to specify the value for configuration paramters in unit specifiers. The unit specifiers are p or P for pages, m or M for megabytes, and g or G for gigabytes. If you do not specify a unit, and you are configuring a parameter that controls memory, Adaptive Server uses the logical page size for the basic unit.

The syntax to indicate a particular unit specification is:

```
sp_configure “parameter name”, 0, “p|P|k|K|m|M|g|G”
```

You must include the “0” as a placeholder.

For example, when configuring max memory for a server that is using 2K pages to 100M, the syntax is:

```
sp_configure “max memory”, 51200
```

However, you can also set max memory for this server to 100M, using the “m” unit specification by typing:

```
sp_configure “max memory”, 0, “100m”
```

You can use this unit specification to configure any parameter. For example, when setting number of locks to 1024 you can enter:

```
sp_configure “number of locks”, 1024
```

or:

```
sp_configure “number of locks”, 0, 1K
```

This functionality will not change the way in which Adaptive Server reports sp_configure output.

Getting help information on configuration parameters

Use either sp_helpconfig or sp_configure to get information on a particular configuration parameter. For example:

```
                            sp_helpconfig "number of open"
                            Configuration option is not unique.
option_name                      config_value run_value
------------------------------ ------------ -----------
number of open databases                 12          12
number of open indexes                  500         500
number of open objects                  500         500

                            sp_helpconfig "number of open indexes"
number of open indexes sets the maximum number of indexes that can be open at
one time on SQL Server. The default value is 500.
Minimum Value Maximum Value Default Value Current Value Memory Used
------------- ------------- ------------- ------------- -----------
          100    2147483647           500           500         208

                            sp_configure "number of open indexes"
Parameter Name           Default  Memory Used  Config Value  Run Value
-----------------------  -------  -----------  ------------  ---------
number of open indexes       500          208           500        500
```

For more information, see "Using sp_helpconfig to get help on configuration parameters" on page 581.

Using *sp_configure*

sp_configure displays and resets configuration parameters. You can restrict the number of parameters displayed by sp_configure using sp_displaylevel to set your display level to one of three values:

- Basic
- Intermediate
- Comprehensive

For information about display levels, see "User-defined subsets of the parameter hierarchy: Display levels" on page 90. For information about sp_displaylevel, see the *Adaptive Server Reference Manual*.

Table 5-1 describes the syntax for sp_configure. The information in the "Effect" column assumes that your display level is set to "comprehensive."

Table 5-1: sp_configure syntax

Command	Effect
sp_configure	Displays all configuration parameters by group, their current values, their default values, the value to which they have most recently been set, and the amount of memory used by this particular setting.
sp_configure "*parameter*"	Displays current value, default value, most recently changed value, and amount of memory used by setting for all parameters matching parameter.
sp_configure "*parameter*", *value*	Resets *parameter* to *value*.
sp_configure "*parameter*", 0, "default"	Resets parameter to its default value.
sp_configure "*group_name*"	Displays all configuration parameters in *group_name*, their current values, their default values, the values to which they were recently set, and the amount of memory used by each setting.
sp_configure "configuration file", 0, "*sub_command*", "*file_name*"	Sets configuration parameters from the configuration file. See "Using sp_configure with a configuration file" on page 83 for descriptions of the parameters.

Syntax elements

In Table 5-1 the following variables are used:

- *parameter* – is any valid Adaptive Server configuration parameter or parameter substring.
- *value* – is any integer within the valid range for that parameter. (See the descriptions of the individual parameters for valid range information.) Parameters that are toggles have only two valid values: 1 (on) and 0 (off).
- *group_name* – is the name of any group in the parameter hierarchy.

Parameter parsing

sp_configure parses each parameter (and parameter name fragment) as "*%parameter%*". A string that does not uniquely identify a particular parameter returns values for all parameters matching the string. For example:

```
sp_configure "lock"
```

returns values for all configuration parameters that include "lock," such as lock shared memory, number of locks, lock promotion HWM, server clock tick length, print deadlock information, and deadlock retries.

Note If you attempt to set a parameter value with a nonunique parameter name fragment, sp_configure returns the current values for all parameters matching the fragment and asks for a unique parameter name.

Using *sp_configure* with a configuration file

You can configure Adaptive Server either interactively, by using sp_configure as described above, or noninteractively, by instructing Adaptive Server to read values from an edited or restored version of the configuration file.

The benefits of using onfiguration files include:

- You can replicate a specific configuration across multiple servers by using the same configuration file.
- You can use a configuration file as a baseline for testing configuration values on your server.
- You can use a configuration file to do validation checking on parameter values before actually setting the values.
- You can create multiple configuration files and switch between them as your resource needs change.

You can make a copy of the configuration file using sp_configure with the parameter "configuration file" and then edit the file at the operating system level. Then, you can use sp_configure with the parameter "configuration file" to instruct Adaptive Server to read values from the edited file. Or you can specify the name of the configuration file at start-up.

For information on editing the file, see "Editing the configuration file" on page 86. For information on specifying the name of the configuration file at start-up, see "Starting Adaptive Server with a configuration file" on page 87.

Naming tips for the configuration file

Each time you modify a configuration parameter with sp_configure, Adaptive Server creates a copy of the outdated configuration file, using the naming convention *server_name.001*, *server_name.002*, *server_name.003*...*server_name.999*.

If you want to work with a configuration file with a name other than the default name, and you keep the *server_name* part of the file name, be sure to include at least one alphabetic character in the extension. Alternatively, you can change the *server_name* part of the file name. Doing this avoids confusion with the backup configuration files generated by Adaptive Server when you modify a parameter.

Using *sp_configure* to read or write the configuration file

The syntax for using the configuration file option with sp_configure is:

```
sp_configure "configuration file", 0, "subcommand", "file_name"
```

where:

- "configuration file" (include quotes) specifies the configuration file parameter.
- 0 must be included as the second parameter to sp_configure for backward compatibility.
- "*subcommand*" is one of the commands described below.
- *file_name* specifies the configuration file you want to use in conjunction with any *subcommand*. If you do not specify a directory as part of the file name, the directory where Adaptive Server was started is used.

Parameters for using configuration files

The four parameters described below can be used with configuration files.

write

write creates *file_name* from the current configuration. If *file_name* already exists, a message is written to the error log; the existing file is renamed using the convention *file_name.001*, *file_name.002*, and so on. If you have changed a static parameter, but you have not restarted your server, write gives you the *currently running value* for that parameter. If you do not specify a directory with *file_name*, the file is written to the directory from which Adaptive Server was started.

read

read performs validation checking on values contained in *file_name* and reads those values that pass validation into the server. If any parameters are missing from *file_name*, the current values for those parameters are used.

If the value of a static parameter in *file_name* is different from its current running value, read fails and a message is printed. However, validation is still performed on the values in *file_name*.

verify

verify performs validation checking on the values in *file_name*. This is useful if you have edited the configuration file, as it prevents you from attempting to configure your server with invalid configuration values.

restore

restore creates *file_name* with the most recently configured values. If you have configured static parameters to new values, this subcommand will write the configured, not the currently running, values to the file. This is useful if all copies of the configuration file have been lost and you need to generate a new copy. If you do not specify a directory with *file_name*, the file is written to the directory from which Adaptive Server was started.

Examples

This example performs validation checking on the values in the file *srv.config* and reads the parameters that pass validation into the server. Current run values are substituted for values that do not pass validation checking.

```
sp_configure "configuration file", 0, "read",
"srv.config"
```

This example creates the file *my_server.config* and writes the current configuration values the server is using to that file.

```
sp_configure "configuration file", 0, "write",
"my_server.config"
```

This example runs validation checking on the values in the file *generic.config*.

```
sp_configure "configuration file", 0, "verify",
"generic.config"
```

This example writes configured values to the file *restore.config*.

```
sp_configure "configuration file", 0, "restore",
"restore.config"
```

Editing the configuration file

The configuration file is an operating system ASCII file that you can edit with any text editor that can save files in ASCII format. The syntax for each parameter is:

```
parameter_name={value | DEFAULT}
```

where *parameter_name* is the name of the parameter you want to specify, *value* is the numeric value for set *parameter_name*, and "DEFAULT" specifies that you want to use the default value for *parameter_name*.

Examples:

```
deadlock retries = 1
```

specifies that the transaction can retry to acquire a lock one time when deadlocking occurs during an index **page split** or shrink.

```
cpu accounting flush interval=DEFAULT
```

specifies that the default value for the parameter cpu accounting flush interval should be used.

When you edit a configuration file, your edits are not validated until you check the file using the verify option, read the file with the read option, or restart Adaptive Server with that configuration file.

If all your configuration files are lost or corrupted, you can re-create one from a running server by using the restore subcommand and specifying a name for the new file. The parameters in the new file will be set to the values with which your server is currently running.

Permissions for configuration files

Configuration files are nonencrypted ASCII text files. By default, they are created with read and write permissions set for the file owner and read permission set for all other users. If you created the configuration file at the operating system level, you are the file owner; if you created the configuration file from Adaptive Server, using the write or restore parameter, the file owner is the user who started Adaptive Server. Usually, this is the user "sybase." To restrict access to configuration files, use your operating system's file permission command to set read, write, and execute permissions as appropriate.

Note You need to set permissions accordingly on *each* configuration file created.

Backing up configuration files

Configuration files are not automatically backed up when you back up the master database. They are operating system files, and you should back them up in the same way you back up your other operating system files.

Checking the name of the configuration file currently in use

The output from sp_configure truncates the name of the configuration file due to space limitations. To see the full name of the configuration file, use:

```
select s1.value2
from syscurconfigs s1, sysconfigures s2
where s1.config = s2.config
and s2.name = "configuration file"
```

Starting Adaptive Server with a configuration file

By default, Adaptive Server reads the configuration file *server_name.cfg* in the start-up directory when it starts. If this file does not exist, it creates a new file and uses Adaptive Server defaults for all values.

You can start Adaptive Server with a specified configuration file. For more information, see the *Utility Guide*.

If the configuration file you specify does not exist, Adaptive Server prints an error message and does not start.

If the command is successful, the file *server_name.bak* is created. This file contains the configuration values stored in sysconfigures prior to the time sysconfigures was updated with the values read in from the configuration file you specified. This file is overwritten with each subsequent start-up.

Configuration file errors

When there are errors in the configuration file, Adaptive Server may not start or may use default values.

Adaptive Server uses default values if:

- There are illegal values. For example, if a parameter requires a numeric value, and the configuration file contains a character string, Adaptive Server uses the default value.
- Values are below the minimum allowable value.

The parameter hierarchy

Configuration parameters are grouped according to the area of Adaptive Server behavior they affect. This makes it easier to identify all parameters that you might need to tune improve a particular area of Adaptive Server performance.

The groups are:

- Backup and recovery
- Cache manager
- Component Integration Services administration
- Disk I/O
- DTM administration
- Error log
- Extended stored procedures
- General information
- Java services
- Languages
- Lock Manager

- Memory use
- Metadata caches
- Network communication
- O/S resources
- Parallel queries
- Physical memory
- Processors
- RepAgent thread administration
- SQL server administration
- Security related
- User environment

Although each parameter has a primary group to which it belongs, many have secondary groups to which they also belong. For example, number of remote connections belongs primarily to the Network Communication group, but it also belongs secondarily to the Adaptive Server Administration group and the Memory Use group. This reflects the fact that some parameters have implications for a number of areas of Adaptive Server behavior. sp_configure displays parameters in all groups to which they belong.

The syntax for displaying all groups and their associated parameters, and the current values for the parameters, is:

```
sp_configure
```

Note The number of parameters sp_configure returns depends on the value to which you have your display level set. See "User-defined subsets of the parameter hierarchy: Display levels" on page 90 for further information about display levels.

The syntax for displaying a particular group and its associated parameter is:

```
sp_configure "group_name"
```

where *group_name* is the name of the group you are interested in. For example, to display the Disk I/O group, type:

```
sp_configure "Disk I/O"
```

```
Group: Disk I/O

Parameter Name           Default Memory Used Config Value Run Value
--------------           ------- ----------- ------------  --------
allow sql server async i/o     1           0             1         1
disk i/o structures          256           0           256       256
number of devices             10           0            10        10
page utilization percent      95           0            95        95
```

Note If the server uses a case-insensitive sort order, sp_configure with no parameters returns a list of all configuration parameters and groups in alphabetical order with no grouping displayed.

User-defined subsets of the parameter hierarchy: Display levels

Depending on your use of Adaptive Server, you may need to adjust some parameters more frequently than others. You may find it is easier to work with a subset of parameters than having to see the entire group when you are working with only a few. You can set your display level to one of three values to give you the subset of parameters that best suits your working style.

The default display level is “comprehensive.” When you set your display level, the setting persists across multiple sessions. However, you can reset it at any time to see more or fewer configuration parameters.

- “Basic” shows just the most basic parameters. It is appropriate for very general server tuning.
- “Intermediate” shows you parameters that are somewhat more complex, in addition to the “basic” parameters. This level is appropriate for a moderately complex level of server tuning.
- “Comprehensive” shows you all the parameters, including the most complex ones. This level is appropriate for users doing highly detailed server tuning.

The syntax for showing your current display level is:

```
sp_displaylevel
```

The syntax for setting your display level is:

```
sp_displaylevel user_name [, basic | intermediate | comprehensive]
```

where *user_name* is your Adaptive Server login name.

The effect of the display level on *sp_configure* output

If your display level is set to either "basic" or "intermediate," sp_configure returns only a subset of the parameters that are returned when your display level is set to "comprehensive." For instance, if your display level is set to "intermediate," and you want to see the parameters in the Languages group, type:

```
sp_configure "Languages"
```

The output would look like this:

```
Group: Languages
Parameter Name                Default Memory Used Config Value Run Value
----------------------------- ------- ----------- ------------ ---------
default character set id            1           0            1         1
default language id                 0           0            0         0
number of languages in cache        3           4            3         3
```

However, this is only a subset of the parameters in the Languages group, because some parameters in that group are displayed only at the "comprehensive" level.

The *reconfigure* command

Pre-11.0 SQL Server releases required you to execute reconfigure after executing sp_configure. Beginning with SQL Server release 11.0, this was no longer required. The reconfigure command still exists, but it does not have any effect. It is included in this release of Adaptive Server so you can run pre-11.0 SQL scripts without modification.

Scripts using reconfigure will still run in the current release, but you should change them at your earliest convenience because reconfigure will not be supported in future releases of Adaptive Server.

Performance tuning with *sp_configure* and *sp_sysmon*

sp_sysmon monitors Adaptive Server performance and generates statistical information that describes the behavior of your Adaptive Server system. See the *Performance and Tuning Guide* for more information.

You can run sp_sysmon before and after using sp_configure to adjust configuration parameters. The output gives you a basis for performance tuning and lets you observe the results of configuration changes.

This chapter includes cross-references to the *Performance and Tuning Guide* for the sp_configure parameters that can affect Adaptive Server performance.

Output from *sp_configure*

The sample output below shows the kind of information sp_configure prints if you have your display level set to "comprehensive" and you execute it with no parameters. The values it prints will vary, depending on your platform and on what values you have already changed.

```
                    sp_configure
Group: General Information
Parameter Name       Default Memory Used Config Value Run Value
--------------       ------- ----------- ------------ ---------
configuration file         0           0            0 /remote/pub

Group: Backup/Recovery

Parameter Name         Default Memory Used Config Value Run Value
--------------         ------- ----------- ------------ ---------
recovery interval in minutes 5           0            5         5
tape retention in days       0           0            0         0
recovery flags               0           0           0        0
...
```

Note All configuration groups and parameters will appear in output if your display level is set to "comprehensive."

The "Default" column displays the value Adaptive Server is shipped with. If you do not explicitly reconfigure a parameter, it retains its default value.

The "Memory Used" column displays the amount of memory used (in kilobytes) by the parameter at its current value. Some related parameters draw from the same memory pool. For instance, the memory used for stack size and stack guard size is already accounted for in the memory used for number of user connections. If you added the memory used by each of these parameters separately, it would total more than the amount actually used. In the "Memory Used" column, parameters that "share" memory with other parameters are marked with a hash mark ("#").

The “Config Value” column displays the most recent value to which the configuration parameter has been set. When you execute sp_configure to modify a dynamic parameter:

- The configuration and run values are updated.
- The configuration file is updated.
- The change takes effect immediately.

When you modify a static parameter:

- The configuration value is updated.
- The configuration file is updated.
- The change takes effect only when you restart Adaptive Server.

The “Run Value” column displays the value Adaptive Server is currently using. It changes when you modify a dynamic parameter’s value with sp_configure and, for static parameters, after you restart Adaptive Server.

The *sysconfigures* and *syscurconfigs* tables

The report displayed by sp_configure is constructed mainly from the master..sysconfigures and master..syscurconfigs system tables, with additional information coming from sysattributes, sysdevices, and other system tables.

The value column in the sysconfigures table records the last value set from sp_configure or the configuration file; the value column in syscurconfigs stores the value currently in use. For dynamic parameters, the two values match; for static parameters, which require a restart of the server to take effect, the two values are different if the values have been changed since Adaptive Server was last started. The values may also be different when the default values are used. In this case, sysconfigures stores 0, and syscurconfigs stores the value that Adaptive Server computes and uses.

sp_configure performs a join on sysconfigures and syscurconfigs to display the values reported by sp_configure.

Querying *syscurconfigs* and *sysconfigures*: An example

You might want to query sysconfigures and syscurconfigs to get information organized the way you want. For example, sp_configure without any arguments lists the memory used for configuration parameters, but it does not list minimum and maximum values. You can query these system tables to get a complete list of current memory usage, as well as minimum, maximum, and default values, with the following query:

```
select b.name, memory_used, minimum_value,
maximum_value, defvalue
from master.dbo.sysconfigures b,
master.dbo.syscurconfigs c
where b.config *= c.config and parent != 19
and b.config > 100
```

Details on configuration parameters

The following sections give both summary and detailed information about each of the configuration parameters. Parameters are listed by group; within each group, they are listed alphabetically.

In many cases, the maximum allowable values for configuration parameters are extremely high. The maximum value for your server is usually limited by available memory, rather than by sp_configure limitations.

Renamed configuration parameters

The following configuration parameters have been renamed:

Old name	New name	See
lock promotion HWM	page lock promotion HWM	“page lock promotion HWM” on page 194
lock promotion LWM	page lock promotion LWM	“page lock promotion LWM” on page 195
lock promotion PCT	page lock promotion PCT	“page lock promotion PCT” on page 196

Replaced configuration parameter

The new lock spinlock ratio parameter replaces the page lock spinlock ration configuration parameter.

Backup and recovery

The following parameters configure Adaptive Server for backing up and recovering data:

number of large i/o buffers

Summary Information	
Name in pre-11.0 version	N/A
Default value	6
Valid values	1–32
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The number of large i/o buffers parameter sets the number of 16K buffers reserved for performing large I/O for certain Adaptive Server utilities. These large I/O buffers are used primarily by the load database command. load database uses one buffer to load the database, regardles of the number of stripes it specifies. load database then uses up to eight buffers to clear the pages for the database it is loading. These buffers are not used by load transaction. If you need to perform more than six load database commands concurrently, configure one large I/O buffer for each load database command.

create database and alter database use these buffers for large I/O while clearing database pages. Each instance of create database or load database can use up to eight large I/O buffers.

These buffers are also used by disk mirroring and by some dbcc commands.

print recovery information

Summary Information	
Name in pre-11.0 release	recovery flags
Default value	0 (off)
Valid values	0 (off), 1 (on)
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

The print recovery information parameter determines what information Adaptive Server displays on the console during recovery. (Recovery is done on each database at Adaptive Server start-up and when a database dump is loaded.) The default value is 0, which means that Adaptive Server displays only the database name and a message saying that recovery is in progress. The other value is 1, which means that Adaptive Server displays information about each individual transaction processed during recovery, including whether it was aborted or committed.

recovery interval in minutes

Summary Information	
Name in pre-11.0 release	recovery interval
Default value	5
Range of values	1–32767
Status	Dynamic
Display level	Basic
Required role	System Administrator

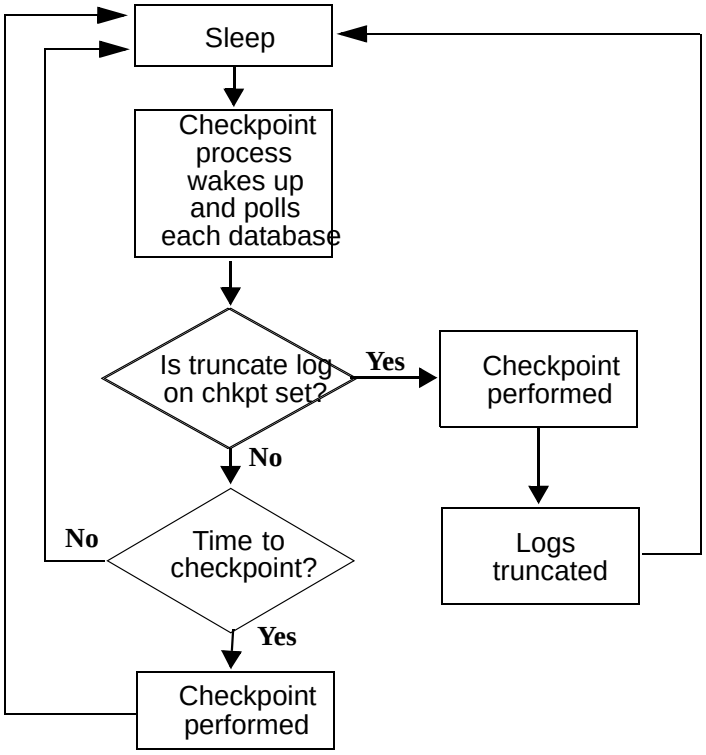
The recovery interval in minutes parameter sets the maximum number of minutes per database that Adaptive Server uses to complete its recovery procedures in case of a system failure. The recovery procedure rolls transactions backward or forward, starting from the transaction that the checkpoint process indicates as the oldest active transaction. The recovery process has more or less work to do depending on the value of recovery interval in minutes.

Adaptive Server estimates that 6000 rows in the transaction log require 1 minute of recovery time. However, different types of log records can take more or less time to recover. If you set recovery interval in minutes to 3, the checkpoint process writes changed pages to disk only when syslogs contains more than 18,000 rows since the last checkpoint.

Note The recovery interval has no effect on long-running, minimally logged transactions (such as create index) that are active at the time Adaptive Server fails. It may take as much time to reverse these transactions as it took to run them. To avoid lengthy delays, dump each database after index maintenance operations.

Adaptive Server uses the recovery interval in minutes setting and the amount of activity on each database to decide when to checkpoint each database. When Adaptive Server checkpoints a database, it writes all **dirty pages** (data pages in cache that have been modified) to disk. This may create a brief period of high I/O, called a *checkpoint spike*.The checkpoint also performs a few other maintenance tasks, including truncating the transaction log for each database for which the truncate log on chkpt option has been set. About once per minute, the sleeping checkpoint process "wakes up," checks the truncate log on chkpt setting, and checks the recovery interval to determine if a checkpoint is needed. Figure 5-1 shows the logic used by Adaptive Server during this process.

Figure 5-1: The checkpoint process



You may want to change the recovery interval if your application and its use change. For example, you may want to shorten the recovery interval when there is an increase in update activity on Adaptive Server. Shortening the recovery interval causes more frequent checkpoints, with smaller, more frequent checkpoint spikes, and slows the system slightly. On the other hand, setting the recovery interval too high might cause the recovery time to be unacceptably long. The spikes caused by checkpointing can be reduced by reconfiguring the housekeeper free write percent parameter. See "housekeeper free write percent" on page 187 for further information. For more information on the performance implications of recovery interval in minutes, see "Tuning the recovery interval" on page 331 in the *Performance and Tuning Guide*.

Use sp_sysmon to determine how a particular recovery interval affects the system. See the *Performance and Tuning Guide* for more information.

tape retention in days

Summary Information	
Name in pre-11.0 release	tape retention
Default value	0
Range of values	0–365
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

The tape retention in days parameter specifies the number of days you intend to retain each tape after it has been used for either a database or a transaction log dump. It is intended to keep you from accidentally overwriting a dump tape.

For example, if you have set tape retention in days to 7 days, and you try to use the tape before 7 days have elapsed since the last time you dumped to that tape, Backup Server issues a warning message.

You can override the warning by using the with init option when executing the dump command. Doing this will cause the tape to be overwritten and all data on the tape to be lost.

Both the dump database and dump transaction commands provide a retaindays option, which overrides the tape retention in days value for a particular dump. See "Protecting dump files from being overwritten" on page 855 for more information.

Cache manager

The parameters in this group configure the data and procedure caches.

global async prefetch limit

Summary Information	
Default value	10
Range of values	0–100
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

The global async prefetch limit parameter specifies the percentage of a buffer pool that can hold the pages brought in by asynchronous prefetch that have not yet been read. This parameter sets the limit for all pools in all caches for which the limit has not been set explicitly with sp_poolconfig.

If the limit for a pool is exceeded, asynchronous prefetch is temporarily disabled until the percentage of unread pages falls below the limit. For more information, see Chapter 25, "Tuning Asynchronous Prefetch," in the *Performance and Tuning Guide*.

global cache partition number

Summary Information	
Default value	1
Range of values	1-64, as powers of 2
Status	Static
Display level	Intermediate
Required role	System Administrator

global cache partition number sets the default number of cache partitions for all data caches. The number of partitions for a particular cache can be set with sp_cacheconfig; the local value takes precedence over the global value.

Use cache partitioning to reduce cache spinlock contention; in general, if spinlock contention exceeds 10%, partitioning the cache should improve performance. Doubling the number of partitions cuts spinlock contention by about one-half.

See "Adding cache partitions" on page 624 for information on configuring cache partitions. See Chapter 25, "Tuning Asynchronous Prefetch" in the *Performance and Tuning Guide* for information.

memory alignment boundary

Summary Information	
Name in pre-11.0 release	calignment
Default value	Logical page size
Range of values	2048[a] – 16384 a. Minimum determined by server's logical page size
Status	Static

Summary Information	
Display level	Comprehensive
Required role	System Administrator

The memory alignment boundary parameter determines the memory address boundary on which data caches are aligned.

Some machines perform I/O more efficiently when structures are aligned on a particular memory address boundary. To preserve this alignment, values for memory alignment boundary should always be powers of two between the logical page size and 2048K.

Note The memory alignment boundary parameter is included for support of certain hardware platforms. Do not modify it unless you are instructed to do so by Sybase Technical Support.

number of index trips

Summary Information	
Name in pre-11.0 release	cindextrips
Default value	0
Range of values	0–65535
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The number of index trips parameter specifies the number of times an aged index page traverses the most recently used/least recently used (MRU/LRU) chain before it is considered for swapping out. As you increase the value of number of index trips, index pages stay in cache for longer periods of time.

A data cache is implemented as an MRU/LRU chain. As the user threads access data and index pages, these pages are placed on the MRU end of the cache's MRU/LRU chain. In some high transaction environments (and in some benchmarks), it is desirable to keep index pages in cache, since they will probably be needed again soon. Setting number of index trips higher keeps index pages in cache longer; setting it lower allows index pages to be swapped out of cache sooner.

You do not need to set the number of index pages parameter for relaxed LRU pages. For more information, see Chapter 19, "Configuring Data Caches."

Note If the cache used by an index is relatively small (especially if it shares space with other objects) and you have a high transaction volume, do not set number of index trips too high. The cache can flood with pages that do not age out, and this may lead to the timing out of processes that are waiting for cache space.

number of oam trips

Summary Information	
Name in pre-11.0 release	coamtrips
Default value	0
Range of values	0–65535
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The number of oam trips parameter specifies the number of times an *object allocation map* (*OAM*) page traverses the MRU/LRU chain before it is considered for swapping out. The higher the value of number of oam trips, the longer aged OAM pages stay in cache.

Each table, and each index on a table, has an OAM page. The OAM page holds information on pages allocated to the table or index and is checked when a new page is needed for the index or table. (See "page utilization percent" on page 112 for further information.) A single OAM page can hold allocation mapping for between 2,000 and 63,750 data or index pages.

The OAM pages point to the allocation page for each allocation unit where the object uses space. The allocation pages, in turn, track the information about extent and page usage within the allocation unit.

In some environments and benchmarks that involve significant allocations of space (that is, massive bulk copy operations), keeping OAM pages in cache longer improves performance. Setting number of oam trips higher keeps OAM pages in cache.

Note If the cache is relatively small and used by a large number of objects, do not set number of oam trips too high. This may result in the cache being flooded with OAM pages that do not age out, and user threads may begin to time out.

procedure cache size

Summary information	
Name in pre-12.5 release	procedure cache percent
Default value	3271
Range of values	3271 – 2147483647
Status	Dynamic
Display level	Basic
Required role	System Administrator

Specifies the size of the procedure cache in 2K pages. Adaptive Server uses the procedure cache while running stored procedures. If the server finds a copy of a procedure already in the cache, it does not need to read it from the disk. Adaptive Server also uses space in the procedure cache to compile queries while creating stored procedures.

Since the optimum value for procedure cache size differs from application to application, resetting it may improve Adaptive Server's performance. For example, if you run many different procedures or ad hoc queries, your application uses the procedure cache more heavily, so you may want to increase this value.

Warning! If procedure cache size is too small, Adaptive Server's performance will be greatly affected.

If you are upgrading

If you are upgrading, procedure cache size is set to the size of the original procedure cache at the time of upgrade. procedure cache size is dynamically configurable, subject to the amount of max memory currently configured.

total data cache size

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0 – 2147483647
Status	Calculated
Display level	Basic
Required role	System Administrator

The total data cache size parameter reports the amount of memory, in kilobytes, that is currently available for data, index, and log pages. It is a calculated value that is not directly user-configurable.

The amount of memory available for the data cache can be affected by a number of factors, including:

- The amount of physical memory available on your machine
- The values to which the following parameters are set:
  - total logical memory
  - number of user connections
  - total procedure cache percent
  - number of open databases
  - number of open objects
  - number of open indexes
  - number of devices

A number of other parameters also affect the amount of available memory, but to a lesser extent.

For information on how Adaptive Server allocates memory and for information on data caches, see "Details on configuration parameters" on page 94.

Component Integration Services administration

The following parameters configure Adaptive Server for Component Integration Services.

cis bulk insert array size

Summary Information	
Name in pre-11.0 release	N/A
Default value	50
Range of values	0–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

When performing a bulk transfer of data from Adaptive Server Enterprise to another Adaptive Server Enterprise, CIS buffers rows internally, and asks the Open Client bulk library to transfer them as a block. The size of the array is controlled by cis bulk insert array size. The default is 50 rows, and the property is dynamic, allowing it to be changed without server reboot

cis bulk insert batch size

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The cis bulk insert batch size parameter determines how many rows from the source table(s) are to be bulk copied into the target table as a single batch using select into.

If the parameter is left at zero (the default), all rows are copied as a single batch. Otherwise, after the count of rows specified by this parameter has been copied to the target table, the server issues a bulk commit to the target server, causing the batch to be committed.

If a normal client-generated bulk copy operation (such as that produced by the bcp utility) is received, then the client is expected to control the size of the bulk batch, and the server ignores the value of this configuration parameter.

cis connect timeout

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0–32767
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The cis connect timeout parameter determines the wait time in seconds for a successful Client-Library connection. By default, no timeout is provided.

cis cursor rows

Summary Information	
Name in pre-11.0 release	N/A
Default value	50
Range of values	1–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The cis cursor rows parameter specifies the cursor row count for cursor open and cursor fetch operations. Increasing this value means more rows will be fetched in one operation. This increases speed but requires more memory. The default is 50.

cis packet size

Summary Information	
Name in pre-11.0 release	N/A

Summary Information	
Default value	512
Range of values	512–32768
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The cis packet size parameter specifies the size of Tabular Data Stream™ (TDS) packets that are exchanged between the server and a remote server when a connection is initiated.

The default packet size on most systems is 512 bytes, and this may be adequate for most applications. However, larger packet sizes may result in significantly improved query performance, especially when text and image or bulk data is involved.

If a packet size larger than the default is specified, and the requested server is a System 10 or later Adaptive Server, then the target server must be configured to allow variable-length packet sizes. Adaptive Server configuration parameters of interest in this case are:

- additional netmem
- maximum network packet size

cis rpc handling

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (off)
Valid values	0 (off), 1 (on)
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The cis rpc handling parameter specifies the default method for remote procedural call (RPC) handling. Setting cis rpc handling to 0 sets the Adaptive Server site handler as the default RPC handling mechanism. Setting the parameter to 1 forces RPC handling to use Component Integration Services access methods. For more information, see the discussion on set cis rpc handling in the *Component Integration Services User's Guide*.

enable cis

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Valid values	0 (off), 1 (on)
Status	Static
Display level	Comprehensive
Required role	System Administrator

The enable cis parameter enables or disables Component Integration Services.

enable file access

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Valid values	0 (off), 1 (on)
Status	Static
Display level	Comprehensive
Required role	System Administrator

Enables access through proxy tables to the External File System. Requires a license for ASE_XFS.

enable full-text search

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Valid values	0 (off), 1 (on)
Status	Static
Display level	Comprehensive
Required role	System Administrator

Enables Enhances Full-Text Search services. Requires a license for ASE_EFTS.

max cis remote connections

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0–2147483647
Status	Dynamic
Display level	Basic
Required role	System Administrator

The max cis remote connections parameter specifies the maximum number of concurrent Client-Library connections that can be made to remote servers by Component Integration Services.

By default, Component Integration Services allows up to four connections per user to be made simultaneously to remote servers. If you set the maximum number of users to 25, up to 100 simultaneous Client-Library connections would be allowed by Component Integration Services.

If this number does not meet the needs of your installation, you can override the setting by specifying exactly how many outgoing Client-Library connections you want the server to be able to make at one time.

Disk I/O

The parameters in this group configure Adaptive Server's disk I/O.

allow sql server async i/o

Summary Information	
Name in pre-11.0 release	T1603 (trace flag)
Default value	1
Valid values	0 (off), 1 (on)
Status	Static
Display level	Comprehensive
Required role	System Administrator

The allow sql server async i/o parameter enables Adaptive Server to run with asynchronous disk I/O. Use asynchronous disk I/O, you have to enable it on *both* Adaptive Server *and* your operating system. See your operating system documentation for information on enabling asynchronous I/O at the operating system level.

In all circumstances, disk I/O runs faster asynchronously than synchronously. This is because when Adaptive Server issues an asynchronous I/O, it does not have to wait for a response before issuing further I/Os.

disable disk mirroring

Summary Information	
Name in pre-11.0 version	N/A
Default value	0
Valid values	1, 0
Status	Static
Display level	Comprehensive
Required role	System Administrator

disable disk mirroring enables or disables disk mirroring for Adaptive Server. This is a global variable; Adaptive Server does not perform any disk mirroring after this configuration parameter is set to 1 and Adaptive Server is rebooted. Setting disable disk mirroring to 0 enables disk mirroring.

Note Disk mirroring must be disabled if you configure Adaptive Server for Failover in a high availability system.

disk i/o structures

Summary Information	
Name in pre-11.0 release	cnblkio
Default value	256
Range of values	0–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The disk i/o structures parameter specifies the initial number of disk I/O control blocks Adaptive Server allocates at start-up.

User processes require a disk I/O control block before Adaptive Server can initiate an I/O request for the process. The memory for disk I/O control blocks is preallocated when Adaptive Server starts. You should configure disk i/o structures to as high a value as your operating system allows, to minimize the chance of running out of disk I/O structures. Refer to your operating system documentation for information on concurrent disk I/Os.

Use sp_sysmon to determine whether you need to allocate more disk I/O structures. See the *Performance and Tuning Guide*. You can set the max asynch i/os per server configuration parameter to the same value as disk i/o structures. See "max async i/os per server" on page 159 for more information.

number of devices

Summary Information	
Name in pre-11.0 release	devices
Default value	10
Range of values	1–256
Status	Dynamic
Display level	Basic
Required role	System Administrator

The number of devices parameter controls the number of database devices Adaptive Server can use. It does not include devices used for database or transaction log dumps.

When you execute disk init, you can also assign the device number (the vdevno). although this value is optional. If you do not assign a vdevno, Adaptive Server assigns the next available virtual device number.

If you do assign a virtual device number, each device number must be unique among the device numbers used by Adaptive Server. The number 0 is reserved for the master device. Legal numbers are 1–256. However, the highest number must be 1 less than the number of database devices you have configured for Adaptive Server. For example, if you configured your server for 10 devices, the legal range of device numbers is 1–9.

To determine which numbers are currently in use, run sp_helpdevice and look in the *device_number* column of output.

If you want to lower the number of devices value after you have added database devices, you must first check to see what device numbers are already in use by database devices. The following command prints the highest value in use:

```
select max(low/power(2,24))+1
    from master..sysdevices
```

Warning! If you set the number of devices value too low in your configuration file, Adaptive Server cannot start. You can find the devices in use by checking the sysdevices system table.

page utilization percent

Summary Information	
Name in pre-11.0 release	N/A
Default value	95
Range of values	1–100
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The page utilization percent parameter is used during page allocations to control whether Adaptive Server scans a table's OAM (*object allocation map*) to find unused pages or simply allocates a new extent to the table. (See "number of oam trips" on page 102 for more information on the OAM.) The page utilization percent parameter is a performance optimization for servers with very large tables; it reduces the time needed to add new space.

If page utilization percent is set to 100, Adaptive Server scans through all OAM pages to find unused pages allocated to the object before allocating a new extent. When this parameter is set lower than 100, Adaptive Server compares the page utilization percent setting to the ratio of used and unused pages allocated to the table, as follows:

```
100 * used pages/(used pages + unused pages)
```

If the page utilization percent setting is lower than the ratio, Adaptive Server allocates a new extent instead of searching for the unused pages.

For example, when inserting data into a 10GB table that has 120 OAM pages and only 1 unused data page:

- A page utilization percent of 100 tells Adaptive Server to scan through all 120 OAM pages to locate an unused data page.
- A page utilization percent of 95 allows Adaptive Server to allocate a new extent to the object, because 95 is lower than the ratio of used pages to used and unused pages.

A low page utilization percent value results in more unused pages. A high page utilization percent value slows page allocations in very large tables, as Adaptive Server performs an OAM scan to locate each unused page before allocating a new extent. This increases logical and physical I/O.

If page allocations (especially in the case of large inserts) seem to be slow, you can lower the value of page utilization percent, but be sure to reset it after inserting the data. A lower setting affects all tables on the server and results in unused pages in all tables.

Fast bulk copy ignores the page utilization percent setting and always allocates new extents until there are no more extents available in the database.

DTM administration

The following parameters configure distributed transaction management (DTM) facilities:

dtm detach timeout period

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (minutes)
Valid values	0 to 2147483647 (minutes)
Status	Dynamic
Display level	10
Required role	System Administrator

dtm detach timeout period sets the amount of time, in minutes, that a distributed transaction branch can remain in the detached state. In some X/Open XA environments, a transaction may become detached from its thread of control (usually to become attached to a different thread of control). Adaptive Server permits transactions to remain in a detached state for the length of time specified by dtm detach timeout period. After this time has passed, Adaptive Server rolls back the detached transaction.

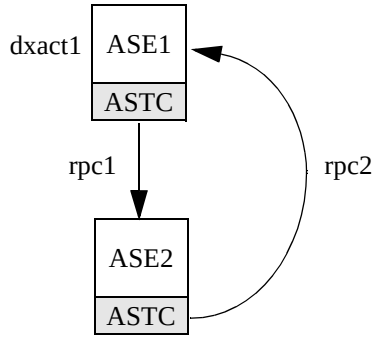
dtm lock timeout period

Summary Information	
Name in pre-11.0 release	N/A
Default value	300 (seconds)
Valid values	1 to 2147483647 (seconds)
Status	Dynamic
Display level	10
Required role	System Administrator

dtm lock timeout period sets the maximum amount of time, in seconds, that a distributed transaction branch will wait for lock resources to become available. After this time has passed, Adaptive Server considers the transaction to be in a deadlock situation, and rolls back the transaction branch that triggered the deadlock. This ultimately rolls back the entire distributed transaction.

Distributed transactions may potentially deadlock themselves if they propagate a transaction to a remote server, and in turn, the remote server propagates a transaction back to the originating server. This situation is shown in Figure 5-2. In Figure 5-2, the work of distributed transaction “dxact1” is propagated to Adaptive Server 2 via “rpc1.” Adaptive Server 2 then propagates the transaction back to the coordinating server via “rpc2.” “rpc2” and “dxact1” share the same gtrid but have different branch qualifiers, so they cannot share the same transaction resources. If “rpc2” is awaiting a lock held by “dxact1”, then a deadlock situation exists.

Figure 5-2: Distributed transaction deadlock



Adaptive Server does not attempt to detect inter-server deadlocks. Instead, it relies on dtm lock timeout period. In Figure 5-2, after dtm lock timeout period has expired, the transaction created for "rpc2" is aborted. This causes Adaptive Server 2 to report a failure in its work, and "dxact1" is ultimately aborted as well.

The value of dtm lock timeout period applies only to distributed transactions. Local transactions may use a lock timeout period with the server-wide lock wait period parameter.

Note Adaptive Server does not use dtm lock timeout period to detect deadlocks on system tables.

enable DTM

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (off)
Valid values	0 (off), 1(on)
Status	Static
Display level	10
Required role	System Administrator

enable DTM enables or disables the Adaptive Server Distributed Transaction Management (DTM) feature. When the DTM feature is enabled, you can use Adaptive Server as a resource manager in X/Open XA and MSDTC systems. You must reboot the server for this parameter to take effect. See the *XA Interface Integration Guide for CICS, Encina, and TUXEDO* for more information about using Adaptive Server in an X/Open XA environment. See Using Adaptive Server Distributed Transaction Management Features for information about transactions in MSDTC environments, and for information aobut Adaptive Server native transaction coordination services.

Note The license information and the Run value for enable DTM are independent of each other. Whether or not you have a license for DTM, the Run value and the Config value are set to 1 after you reboot Adaptive Server. And until you have a license, you cannot run DTM. If you have not installed a valid license, Adaptive Server logs an error message and does not activate the feature. See your Installation Guide for information about installing license keys.

The lisence information and the configuration value are indepent of each other.

enable xact coordination

Summary Information	
Name in pre-11.0 release	N/A
Default value	1 (on)
Valid values	0 (off), 1(on)
Status	Static
Display level	10
Required role	System Administrator

enable xact coordination enables or disables Adaptive Server transaction coordination services. When this parameter is set to 1 (on), coordination services are enabled, and the server can propagate transactions to other Adaptive Servers. This may occur when a transaction executes a remote procedure call (RPC) to update data in another server, or updates data in another server using Component Integration Services (CIS). Transaction coordination services ensure that updates to remote Adaptive Server data commit or roll back with the original transaction.

If this parameter is set to 0 (off), Adaptive Server will not coordinate the work of remote servers. Transactions can still execute RPCs and update data using CIS, but Adaptive Server cannot ensure that remote transactions are rolled back with the original transaction or that remote work is committed along with an original transaction, if remote servers experience a system failure. This corresponds to the behavior of Adaptive Server versions prior to version 12.x.

number of dtx participants

Summary Information	
Name in pre-11.0 release	N/A
Default value	500
Valid values	100 to 2147483647
Status	Dynamic
Display level	10
Required role	System Administrator

number of dtx participants sets the total number of remote transactions that the Adaptive Server transaction coordination service can propagate and coordinate at one time. A DTX participant is an internal memory structure that the coordination service uses to manage a remote transaction branch. As transactions are propagated to remote servers, the coordination service must obtain new DTX participants to manage those branches.

By default, Adaptive Server can coordinate 500 remote transactions. Setting number of dtx participants to a smaller number reduces the number of remote transactions that the server can manage. If no DTX participants are available, new distributed transactions will be unable to start. In-progress distributed transactions may abort if no DTX participants are available to propagate a new remote transaction.

Setting number of dtx participants to a larger number increases the number of remote transaction branches that Adaptive Server can handle, but also consumes more memory.

Optimizing the number of dtx participants for Your System

During a peak period, use sp_monitorconfig to examine the use of DTX participants:

```
                        sp_monitorconfig "number of dtx participants"
Usage information at date and time: Jun 18 1999 9:00AM.
Name            # Free   # Active   % Active  # Max Ever Used   Re-used
--------------  --------  ------   --------   ---------------   -------
number of dtx   480          20         4.00             210         NA
participants
```

If the #Free value is zero or very low, new distributed transactions may be unable to start due to a lack of DTX participants. Consider increasing the number of dtx participants value.

If the #Max Ever Used value is too low, unused DTX participants may be consuming memory that could be used by other server functions. Consider reducing the value of number of dtx participants.

strict dtm enforcement

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (off)
Valid values	0 (off), 1(on)
Status	Static
Display level	10
Required role	System Administrator

strict dtm enforcement determines whether or not Adaptive Server transaction coordination services will strictly enforce the ACID properties of distributed transactions.

In environments where Adaptive Server should propagate and coordinate transactions only to other Adaptive Servers that support transaction coordination, set strict dtm enforcement to 1 (on). This ensures that transactions are propagated only to servers that can participate in Adaptive Server-coordinated transactions, and transactions complete in a consistent manner. If a transaction attempts to update data in a server that does not support transaction coordination services, Adaptive Server aborts the transaction.

In heterogeneous environments, you may want to make use of servers that do not support transaction coordination. This includes older versions of Adaptive Server and non-Sybase database stores configured using CIS. Under these circumstances, you can set strict dtm enforcement to 0 (off). This allows Adaptive Server to propagate transactions to legacy Adaptive Servers and other data stores, but does not ensure that the remote work of these servers is rolled back or committed with the original transaction.

txn to pss ratio

Summary Information	
Name in pre-11.0 release	N/A
Default value	16
Valid values	1 to 2147483647
Status	Static
Display level	1
Required role	System Administrator

Adaptive Server manages transactions as configurable server resources. Each time a new transaction begins, Adaptive Server must obtain a free **transaction descriptor** from a global pool that is created at boot time. Transaction descriptors are internal memory structures that Adaptive Server uses to represent active transactions.

Adaptive Server requires one free transaction descriptor for:

- The outer block of each server transaction. The outer block of a transaction may be created explicitly when a client executes a new begin transaction command. Adaptive Server may also implicitly create an outer transaction block when clients use Transact-SQL to modify data without using begin transaction to define the transaction.

  Note Subsequent, nested transaction blocks, created with additional begin transaction commands, do not require additional transaction descriptors.

- Each database accessed in a **multi-database transaction**. Adaptive Server must obtain a new transaction descriptor each time a transaction uses or modifies data in a new database.

txn to pss ratio determines the total number of transaction descriptors available to the server. At boot time, this ratio is multiplied by the number of user connections parameter to create the transaction descriptor pool:

```
# of transaction descriptors = number of user connections * txn to pss ratio
```

The default value, 16, ensures compatibility with earlier versions of Adaptive Server. Prior to version 12.x, Adaptive Server allocated 16 transaction descriptors for each user connection. In version 12.x, the number of simultaneous transactions is limited only by the number of transaction descriptors available in the server.

Note The number of databases accessed in a multi-database transaction remains limited to 16.

Optimizing the txn to pss ratio for Your System

During a peak period, use sp_monitorconfig to examine the use of transaction descriptors:

```
                    sp_monitorconfig "txn to pss ratio"
Usage information at date and time: Jun 18 1999 8:54AM.
Name            # Free   # Active   % Active  # Max Ever Used   Re-used
--------------  --------  ------   --------   ---------------   -------
txn to pss       784          80       10.20         523           NA
ratio
```

If the #Free value is zero or very low, transactions may be delayed as Adaptive Server waits for transaction descriptors to become free in the server. In this case, you should consider increasing the value of txn to pss ratio.

If the #Max Ever Used value is too low, unused transaction descriptors may be consuming memory that can be used by other server functions. Consider reducing the value of txn to pss ratio.

xact coordination interval

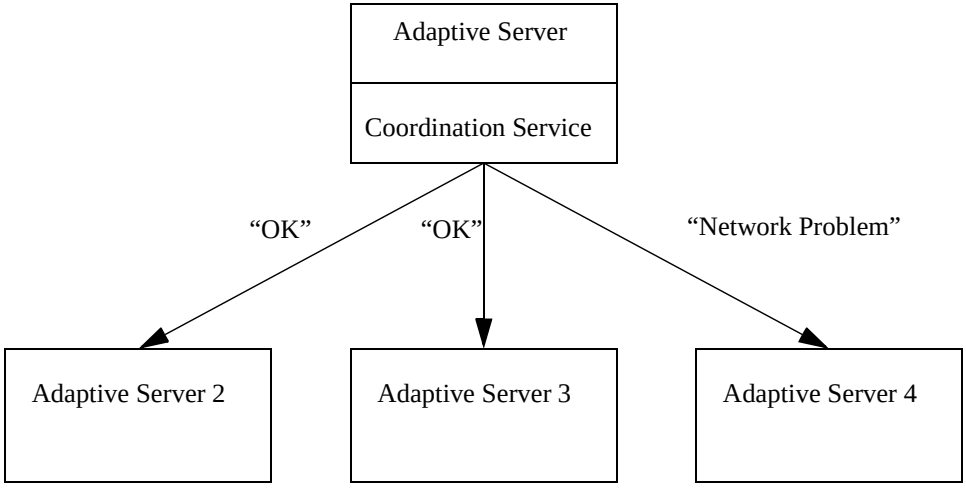
Summary Information	
Name in pre-11.0 release	N/A
Default value	60 (seconds)
Valid values	1 to 2147483647 (seconds)
Status	Dynamic

Summary Information	
Display level	10
Required role	System Administrator

xact coordination interval defines the length of time between attempts to resolve transaction branches that were propagated to remote servers.

The coordinating Adaptive Server makes regular attempts to resolve the work of remote servers participating in a distributed transaction. The coordinating server contacts each remote server participating in the distributed transaction in a serial manner, as shown in Figure 5-3. The coordination service may be unable to resolve a transaction branch for a variety of reasons. For example, if the remote server is not reachable due to network problems, the coordinating server reattempts the connection after the time specified by xact coordination level.

Figure 5-3: Resolving remote transaction branches



With the default value of xact coordination interval, 60, Adaptive Server attempts to resolve remote transactions once every minute. Decreasing the value may speed the completion of distributed transactions, but only if the transactions are themselves resolved in less than a minute. Under normal circumstances, there is no performance penalty to decreasing the value of xact coordination interval.

Setting xact coordination interval to a higher number can slow the completion of distributed transactions, and cause transaction branches to hold resources longer than they normally would. Under normal circumstances, you should not increase the value of xact coordination interval beyond its default.

Error log

The parameters in this group configure the Adaptive Server error log and the logging of Adaptive Server events to the Windows NT Event Log.

event log computer name (Windows NT only)

Summary Information	
Name in pre-11.0 release	N/A
Default value	'LocalSystem'
Valid values	• Name of an NT machine on the network configured to record Adaptive Server messages • 'LocalSystem' • 'NULL'
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The event log computer name parameter specifies the name of the Windows NT PC that logs Adaptive Server messages in its Windows NT Event Log. You can use this parameter to have Adaptive Server messages logged to a remote machine. This feature is available on Windows NT servers only.

A value of 'LocalSystem' or 'NULL' specifies the default local system.

You can also use the Server Config utility to set the event log computer name parameter by specifying the Event Log Computer Name under Event Logging.

Setting the event log computer name parameter with sp_configure or specifying the Event Log Computer Name under Event Logging overwrites the effects of the command line -G option, if it was specified. If Adaptive Server was started with the -G option, you can change the destination remote machine by setting the event log computer name parameter.

For more information about logging Adaptive Server messages to a remote site, see *Configuring Adaptive Server for Windows NT*.

event logging (Windows NT only)

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Valid values	0 (off), 1 (on)
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The event logging parameter enables and disables the logging of Adaptive Server messages in the Windows NT Event Log. This feature is available on Windows NT servers only.

The default value of 1 enables Adaptive Server message logging in the Windows NT Event Log; a value of 0 disables it.

You use the Server Config utility to set the event logging parameter by selecting "Use Windows NT Event Logging" under Event Logging.

Setting the event logging parameter or selecting "Use Windows NT Event Logging" overwrites the effects of the command line -g option, if it was specified.

log audit logon failure

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (off)
Range of values	0 (off), 1 (on)
Status	Dynamic

Summary Information	
Display level	Comprehensive
Required role	System Administrator

The log audit logon failure parameter specifies whether to log unsuccessful Adaptive Server logins to the Adaptive Server error log and, on Windows NT servers, to the Windows NT Event Log, if event logging is enabled.

A value of 1 requests logging of unsuccessful logins; a value of 0 specifies no logging.

log audit logon success

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (off)
Range of values	0 (off), 1 (on)
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The log audit logon success parameter specifies whether to log successful Adaptive Server logins to the Adaptive Server error log and, on Windows NT servers, to the Windows NT Event Log, if event logging is enabled.

A value of 1 requests logging of successful logins; a value of 0 specifies no logging.

Extended stored procedures

The parameters in this group affect the behavior of extended stored procedures (ESPs).

esp execution priority

Summary Information	
Name in pre-11.0 release	N/A
Default value	8
Range of values	0–15

Summary Information	
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The esp execution priority parameter sets the priority of the XP Server thread for ESP execution. ESPs can be CPU-intensive over long periods of time. Also, since XP Server resides on the same machine as Adaptive Server, XP Server can impact Adaptive Server's performance.

Use esp execution priority to set the priority of the XP Server thread for ESP execution. See the *Open Server Server-Library/C Reference Manual* for information about scheduling Open Server threads.

esp execution stacksize

Summary Information	
Name in pre-11.0 release	N/A
Default value	34816
Range of values	34816–2^{14}
Status	Static
Display level	Comprehensive
Required role	System Administrator

The esp execution stacksize parameter sets the size of the stack, in bytes, to be allocated for ESP execution.

Use this parameter if you have your own ESP functions that require a larger stack size than the default, 34816.

esp unload dll

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (off)
Range of values	0 (off), 1 (on)
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The esp unload dll parameter specifies whether DLLs that support ESPs should be automatically unloaded from XP Server memory after the ESP call has completed.

If esp unload dll is set to 0, DLLs are not automatically unloaded. If it is set to 1, they are automatically unloaded.

If esp unload dll is set to 0, you can still unload individual DLLs explicitly at runtime, using sp_freedll.

start mail session (Windows NT only)

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (off)
Valid values	0 (off), 1 (on)
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The start mail session parameter enables and disables the automatic initiation of an Adaptive Server mail session when you start Adaptive Server. This feature is available on Windows NT servers only.

A value of 1 configures Adaptive Server to start a mail session the next time Adaptive Server is started. A value of 0 configures Adaptive Server not to start a mail session at the next restart.

If start mail session is 0, you can start an Adaptive Server mail session explicitly, using the xp_startmail system ESP.

Before setting the start mail session parameter, you must prepare your Windows NT system by creating a mailbox and mail profile for Adaptive Server. Then, you must create an Adaptive Server account for Sybmail. See *Configuring Adaptive Server for Windows NT* for information about preparing your system for Sybmail.

xp_cmdshell context

Summary Information	
Name in pre-11.0 release	N/A
Default value	1

Summary Information	
Valid values	0, 1
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The xp_cmdshell context parameter sets the security context for the operating system command to be executed using the xp_cmdshell system ESP.

Setting xp_cmdshell context to 1 restricts the xp_cmdshell security context to users who have accounts at the operating system level. Its behavior is platform-specific. If xp_cmdshell context is set to 1, to use an xp_cmdshell ESP, an operating system user account must exist for the Adaptive Server user name. For example, an Adaptive Server user named “sa” will not be able to use xp_cmdshell unless he or she has an operating system level user account named “sa”.

On Windows NT, when xp_cmdshell context is set to 1, xp_cmdshell succeeds only if the user name of the user logging in to Adaptive Server is a valid Windows NT user name with Windows NT system administration privileges on the system on which Adaptive Server is running.

On other platforms, when xp_cmdshell context is set to 1, xp_cmdshell succeeds only if Adaptive Server was started by a user with “superuser” privileges at the operating system level. When Adaptive Server gets a request to execute xp_cmdshell, it checks the *uid* of the user name of the ESP requestor and runs the operating system command with the permissions of that *uid.*

If xp_cmdshell context is 0, the permissions of the operating system account under which Adaptive Server is running are the permissions used to execute an operating system command from xp_cmdshell. This allows users to execute operating commands that they would not ordinarily be able to execute under the security context of their own operating system accounts.

General information

The parameter in this group is not related to any particular area of Adaptive Server behavior.

configuration file

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	N/A
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The configuration file parameter specifies the location of the configuration file currently in use. See "Using sp_configure with a configuration file" on page 83 for a complete description of configuration files.

In sp_configure output, the "Run Value" column displays only 10 characters. For this reason, the output may not display the entire path and name of your configuration file.

Java services

The parameters in this group enable and configure memory for Java in Adaptive Server. See the *Java in Adaptive Server Enterprise* manual for complete information about Java in the database.

If you use method calls to JDBC, you may need to increase the size of the execution stack available to the user. See "stack size" on page 229 for information about setting the stack size parameter.

enable java

Summary information	
Name in pre-11.0 release	N/A
Default value	0 (disabled)
Range of values	0 (disabled), 1 (enabled)
Status	Static
Display level	Comprehensive
Required role	System Administrator

The enable java parameter enables and disables Java in the Adaptive Server database. You cannot install Java classes or perform any Java operations until the server is enabled for Java.

Note The license information and the Run value for enable java are independent of each other. Whether or not you have a license for java, the Run value and the Config value are set to 1 after you reboot Adaptive Server. You cannot run Java until you have a license. If you have not installed a valid license, Adaptive Server logs an error message and does not activate the feature. See your installation guide for information about installing license keys.

enable enterprise java beans

Summary information	
Name in pre-11.0 release	N/A
Default value	0 (disabled)
Range of values	0 (disabled), 1 (enabled)
Status	Static
Display level	Comprehensive
Required role	System Administrator

The enable enterprise java beans parameter enables and disables EJB Server in the Adaptive Server database. You cannot use EJB Server until the Adaptive Server is enabled for EJB Server.

Note The license information and the Run value for enable java beans are independent of each other. Whether or not you have a license for java, the Run value and the Config value are set to 1 after you reboot Adaptive Server. You cannot run EJB Server until you have a license. If you have not installed a valid license, Adaptive Server logs an error message and does not activate the feature. See your installation guide for information about installing license keys.

size of global fixed heap

Summary information	
Name in pre-11.0 release	N/A
Default values	150 pages (32-bit version) 300 pages (64-bit version)
Minimum values	10 pages (32-bit version) 20 pages (64-bit version)
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The size of global fixed heap parameter specifies the memory space for internal data structures and other needs.

If you change the size of the global fixed heap, you must also change the total logical memory by the same amount.

size of process object heap

Summary information	
Name in pre-11.0 release	N/A
Default values	1500 pages (32-bit version) 3000 pages (64-bit version)
Minimum values	45 pages (32-bit version) 90 pages (64-bit version)
Status	Dynamic
Display level	Basic
Required role	System Administrator

The size of process object fixed heap parameter specifies the total memory space for all processes using the Java VM.

If you change the size of the process object fixed heap, you must change the total logical memory by that amount.

size of shared class heap

Summary information	
Name in pre-11.0 release	N/A

Summary information	
Default values	1536 pages (32-bit version) 3072 pages (64-bit version)
Minimum values	650 pages (32-bit version) 1300 pages (64-bit version)
Status	Dynamic
Display level	Basic
Required role	System Administrator

The size of shared class heap parameter specifies the shared memory space for all Java classes called into the Java VM. Adaptive Server maintains the shared class heap server-wide for both user-defined and system-provided Java classes.

If you change the size of the shared class heap, you must change the total logical memory by the same amount.

Languages

The parameters in this group configure languages, sort orders, and character sets.

default character set id

Summary Information	
Name in pre-11.0 release	default character set id
Default value	1
Range of values	0–255
Status	Static
Display level	Intermediate
Required role	System Administrator

The default character set id parameter specifies the number of the default character set used by the server. The default is set at installation time, and can be changed later with the Sybase installation utilities. See Chapter 7, "Configuring Character Sets, Sort Orders, and Languages," for a discussion of how to change character sets and sort orders.

default language id

Summary Information	
Name in pre-11.0 release	default language
Default value	0
Range of values	0–32767
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

The default language id parameter is the number of the language that is used to display system messages unless a user has chosen another language from those available on the server. us_english always has an ID of NULL. Additional languages are assigned unique numbers as they are added.

default sortorder id

Summary Information	
Name in pre-11.0 release	default sortorder id
Default value	50
Range of values	0–255
Status	Static
Display level	Comprehensive
Required role	System Administrator

The default sortorder id parameter is the number of the sort order that is installed as the default on the server. To change the default sort order, see Chapter 7, “Configuring Character Sets, Sort Orders, and Languages.”

disable character set conversions

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (enabled)
Valid values	0 (enabled), 1 (disabled)
Status	Static
Display level	Comprehensive
Required role	System Administrator

Changing disable character set conversions to 1 turns off character set conversion for data moving between clients and Adaptive Server. By default, Adaptive Server performs conversion on data moving to and from clients that use character sets that are different than the server's. For example, if some clients use Latin-1 (iso_1) and Adaptive Server uses Roman-8 (roman8) as its default character set, data from the clients is converted to Roman-8 when being loaded into Adaptive Server. For clients using Latin-1, the data is reconverted when it is sent to the client; for clients using the same character set as Adaptive Server, the data is not converted.

By setting disable character set conversions, you can request that no conversion take place. For example, if all clients are using a given character set, and you want Adaptive Server to store all data in that character set, you can set disable character set conversions to 1, and no conversion will take place.

Lock Manager

The parameters in this group configure locks.

lock address spinlock ratio

Summary Information	
Name in pre-11.0 release	N/A
Default value	100
Range of values	1–2147483647
Status	Static
Display level	Comprehensive
Required role	System Administrator

For Adaptive Servers running with multiple engines, the address lock spinlock ratio sets the number of rows in the internal address locks hash table that are protected by one spinlock.

Adaptive Server manages the acquiring and releasing of address locks using an internal hash table with 1031 rows (known as hash buckets). This table can use one or more spinlocks to serialize access between processes running on different engines.

Adaptive Server's default value for address lock spinlock ratio is 100, which defines 11 spinlocks for the address locks hash table. The first 10 spinlocks protect 100 rows each, and the eleventh spinlock protects the remaining 31 rows. If you specify a value of 1031 or greater for address lock spinlock ratio, Adaptive Server uses only 1 spinlock for the entire table.

number of locks

Summary Information	
Name in pre-11.0 release	locks
Default value	5000
Range of values	1000–2147483647
Status	Dynamic
Display level	Basic
Required role	System Administrator

The number of locks parameter sets the total number of available locks for all users on Adaptive Server.

The total number of locks needed by Adaptive Server depends on the number and nature of the queries that are running. The number of locks required by a query can vary widely, depending on the number of concurrent and parallel processes and the types of actions performed by the transactions. To see how many locks are in use at a particular time, use sp_lock.

For serial operation, we suggest that you can start with an arbitrary number of 20 locks for each active, concurrent connection.

Parallel execution requires more locks than serial execution. For example, if you find that queries use an average of five worker processes, try increasing, by one-third, the number of locks configured for serial operation.

If the system runs out of locks, Adaptive Server displays a server-level error message. If users report lock errors, it typically indicates that you need to increase number of locks; but remember that locks use memory. See "Number of locks" on page 589 for information.

Note Datarows locking may require that you change the value for number of locks. See the Performance and Tuning Guide. for more information.

deadlock checking period

Summary Information	
Name in pre-11.0 release	N/A
Default value	500
Range of values	0–2147483
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

deadlock checking period specifies the minimum amount of time (in milliseconds) before Adaptive Server initiates a deadlock check for a process that is waiting on a lock to be released. Deadlock checking is time-consuming overhead for applications that experience no deadlocks or very few, and the overhead grows as the percentage of lock requests that must wait for a lock also increases.

If you set this value to a nonzero value (*n*), Adaptive Server initiates a deadlock check after a process waits at least *n* milliseconds. For example, you can make a process wait at least 700 milliseconds. for a lock before each deadlock check as follows:

```
sp_configure "deadlock checking period", 700
```

If you set this parameter to 0, Adaptive Server initiates deadlock checking when each process begins to wait for a lock. Any value less than the number of milliseconds in a clock tick is treated as 0. See "sql server clock tick length" on page 207 for more information.

Configuring deadlock checking period to a higher value produces longer delays before deadlocks are detected. However, since Adaptive Server grants most lock requests before this time elapses, the deadlock checking overhead is avoided for those lock requests. If your applications deadlock infrequently, set deadlock checking period to a higher value to avoid the overhead of deadlock checking for most processes. Otherwise, the default value of 500 should suffice.

Use sp_sysmon to determine the frequency of deadlocks in your system and the best setting for deadlock checking period. See the *Performance and Tuning Guide* for more information.

deadlock retries

Summary Information	
Name in pre-11.0 release	N/A
Default value	5
Range of values	0–2147483647
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

deadlock retries specifies the number of times a transaction can attempt to acquire a lock when deadlocking occurs during an index **page split** or shrink.

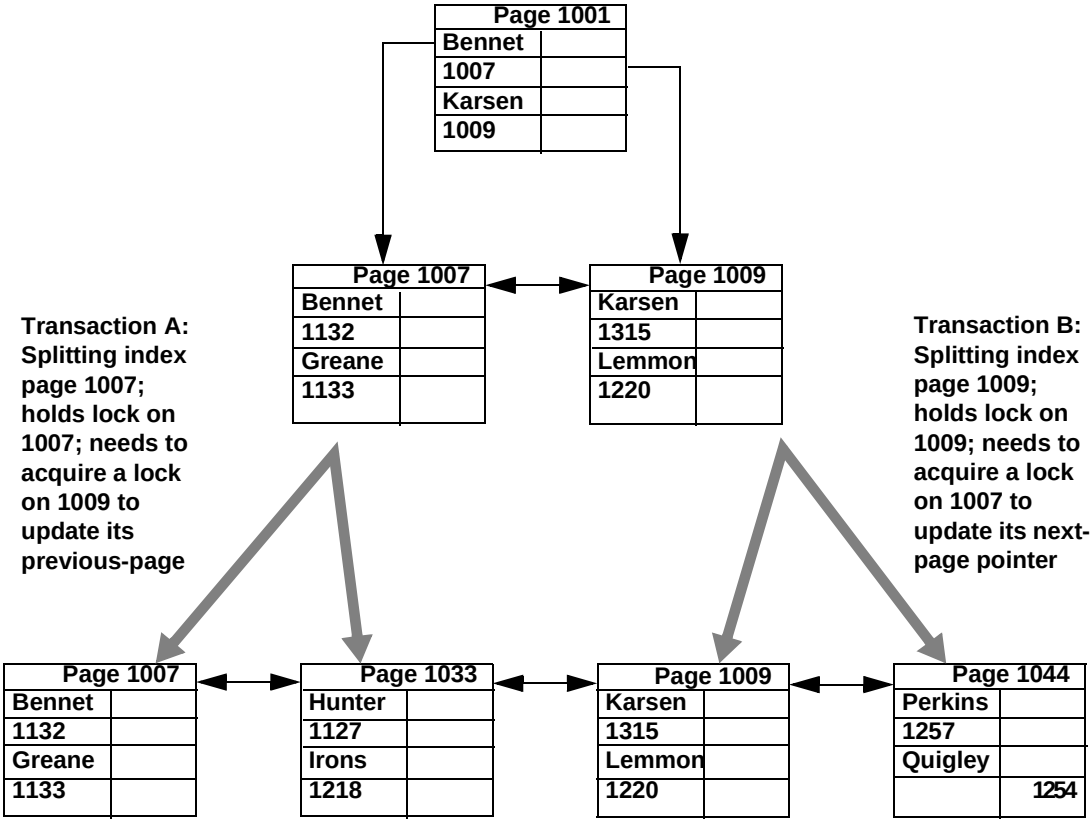
For example, Figure 5-4 illustrates the following scenario:

- Transaction A locks page 1007 and needs to acquire a lock on page 1009 to update the page pointers for a page split.
- Transaction B is also inserting an index row that causes a page split, holds a lock on page 1009, and needs to acquire a lock on page 1007.

In this situation, rather than immediately choosing a process as a deadlock victim, Adaptive Server relinquishes the index locks for one of the transactions. This often allows the other transaction to complete and release its locks.

For the transaction that surrendered its locking attempt, the index is rescanned from the root page, and the page split operation is attempted again, up to the number of times specified by deadlock retries.

Figure 5-4: Deadlocks during page splitting in a clustered index



sp_sysmon reports on deadlocks and retries. See the *Performance and Tuning Guide* for more information.

lock spinlock ratio

Summary Information	
Default value	85
Range of values	1–2147483647
Status	Static
Display level	Comprehensive
Required role	System Administrator

Adaptive Server manages the acquiring and releasing of locks using an internal hash table with a configurable number of hash buckets. On SMP systems, this hash table can use one or more spinlocks to serialize access between processes running on different engines. To set the number of hash buckets, use the lock hashtable size.

For Adaptive Servers running with multiple engines, the lock spinlock ratio sets a ratio that determines the number of lock hash buckets that are protected by one spinlock. If you increase lock hashtable size, the number of spinlocks increases, so the number of hash buckets protected by one spinlock remains the same.

Adaptive Server's default value for lock spinlock ratio is 85. With lock hashtable size set to the default value of 2048, the default spinlock ratio defines 26 spinlocks for the lock hash table. For more information about configuring spinlock ratios, see "Configuring spinlock ratio parameters" on page 644 of the *System Administration Guide*.

sp_sysmon reports on the average length of the hash chains in the lock hash table. See the Performance and Tuning Guide for more information.

lock hashtable size

Summary Information	
Default value	2048
Range of values	1–2147483647
Status	Static
Display level	Comprehensive
Required role	System Administrator

lock hashtable size specifies the number of *hash buckets* in the lock hash table. This table manages all row, page, and table locks and all lock requests. Each time a task acquires a lock, the lock is assigned to a hash bucket, and each lock request for that lock checks the same hash bucket. Setting this value too low results in large numbers of locks in each hash bucket and slows the searches. On Adaptive Servers with multiple engines, setting this value too low can also lead to increased spinlock contention. Do not set the value to less than the default value, 2048.

lock hashtable size must be a power of 2. If the value you specify is not a power of 2, sp_configure rounds the value to the next highest power of 2 and prints an informational message.

The optimal hash table size is a function of the number of distinct objects (pages, tables, and rows) that will be locked concurrently. The optimal hash table size is at least 20 percent of the number of distinct objects that need to be locked concurrently. See Performance and Tuning Guide for more information on configuring the lock hash table size.

lock scheme

Summary Information	
Default value	allpages
Range of values	allpages, datapages, datarows
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

lock scheme sets the default locking scheme to be used by create table and select into commands when a lock scheme is not specified in the command.

The values for lock scheme are character data, so you must use 0 as a placeholder for the second parameter, which must be numeric, and specify allpages, datapages, or datarows as the third parameter:

```
sp_configure "lock scheme", 0, datapages
```

lock wait period

Summary Information	
Default value	2147483647
Range of values	0–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

lock wait period limits the number of seconds that tasks wait to acquire a lock on a table, data page, or data row. If the task does not acquire the lock within the specified time period, Adaptive Server returns error message 12205 to the user and rolls back the transaction.

The lock wait option of the set command sets a session-level number of seconds that a task will wait for a lock. It overrides the server-level setting for the session.

At the default value, all processes wait indefinitely for locks. To restore the default value, reset the value to 2147483647 or use:

```
sp_configure "lock wait period", 0, "default"
```

read committed with lock

Summary Information	
Default value	0 (off)
Valid values	0 (off), 1(on)
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

read committed with lock determines whether an Adaptive Server using transaction isolation level 1 (read committed) holds shared locks on rows or pages of data-only-locked tables during select queries. For cursors, the option applies only to cursors declared as read-only. By default, this parameter is turned off to reduce lock contention and blocking. This parameter affects only queries on data-only locked tables.

For transaction isolation level 1, select queries on allpages-locked tables continue to hold locks on the page at the current position. Any updatable cursor on a data-only-locked table also holds locks on the current page or row. See the Performance and Tuning Guide for more information.

lock table spinlock ratio

Summary Information	
Name in pre-11.0 release	N/A
Default value	20
Range of values	1–2147483647
Status	Static
Display level	Comprehensive
Required role	System Administrator

For Adaptive Servers running with multiple engines, the table lock spinlock ratio configuration parameter sets the number of rows in the internal table locks hash table that are protected by one **spinlock**.

Adaptive Server manages the acquiring and releasing of table locks using an internal hash table with 101 rows (known as hash buckets). This table can use one or more spinlocks to serialize access between processes running on different engines.

Adaptive Server's default value for table lock spinlock ratio is 20, which defines 6 spinlocks for the table locks hash table. The first 5 spinlocks protect 20 rows each; the sixth spinlock protects the last row. If you specify a value of 101 or greater for table lock spinlock ratio, Adaptive Server uses only 1 spinlock for the entire table.

Memory use

The following parameter optimizes Adaptive Server's memory use:

executable codesize + overhead

Summary Information	
Name in pre-11.0 release	sql server code size
Default value	0
Range of values	0-2147483647
Status	Calculated
Display level	Basic
Required role	System Administrator

executable codesize + overhead reports the combined size (in kilobytes) of the Adaptive Server executable and overhead. It is a calculated value and is not user-configurable.

Metadata caches

The following parameters help set the metadata cache size for frequently used system catalog information. The *metadata cache* is a reserved area of memory used for tracking information on databases, indexes, or objects. The greater the number of open databases, indexes, or objects, the larger the metadata cache size. For a discussion of metadata caches in a memory-usage context, see "Open databases, open indexes, and open objects" on page 588.

number of open databases

Summary Information	
Name in pre-11.0 release	open databases
Default value	12
Range of values	5–2147483647
Status	Dynamic
Display level	Basic
Required role	System Administrator

number of open databases sets the maximum number of databases that can be open simultaneously on Adaptive Server.

When you calculate a value, include the system databases master, model, sybsystemprocs, and tempdb. If you have installed auditing, include the sybsecurity database. Also, count the sample databases pubs2 and pubs3, the syntax database sybsyntax, and the dbcc database dbccdb if they are installed.

If you are planning to make a substantial change, such as loading a large database from another server, you can calculate an estimated metadata cache size by using sp_helpconfig. sp_helpconfig displays the amount of memory required for a given number of metadata descriptors, as well as the number of descriptors that can be accommodated by a given amount of memory. A database metadata descriptor represents the state of the database while it is in use or cached between uses.

Optimizing the *number of open databases* parameter for your system

If Adaptive Server displays a message saying that you have exceeded the allowable number of open databases, you will need to adjust the value.

To set the number of open databases parameter optimally:

- Step 1: Determine the total number of databases (database metadata descriptors).
- Step 2: Reset number of open databases to that number.
- Step 3: Find the number of active databases (active metadata descriptors) during a peak period.
- Step 4: Reset number of open databases to that number, plus 10 percent.

The following section details the basic steps listed above.

1 Use the sp_countmetadata system procedure to find the total number of database metadata descriptors. For example:

```
sp_countmetadata "open databases"
```

The best time to run sp_countmetadata is when there is little activity on the server. Running sp_countmetadata during a peak time can cause contention with other processes.

Suppose Adaptive Server reports the following information:

```
There are 50 databases, requiring 1719 Kbytes of
memory. The 'open databases' configuration
parameter is currently set to 500.
```

2 Configure number of open databases with the value of 50:

```
sp_configure "number of open databases", 50
```

This new configuration is only a start; the ideal size should be based on the number of *active* metadata database cache descriptors, not the *total* number of databases.

3 During a peak period, find the number of active metadata descriptors. For example:

```
sp_monitorconfig "open databases"
Usage information at date and time: Jan 14 1997  8:54AM.
Name            # Free   # Active   % Active  # Max Ever Used   Re-used
--------------  --------  ------   --------   ---------------   -------
number of open  50        20       40.00      26                No
databases
```

At this peak period, 20 metadata database descriptors are active; the maximum number of descriptors that have been active since the server was last started is 26.

See sp_monitorconfig in the *Adaptive Server Reference Manual* for more information.

4 Configure number of open databases to 26, plus additional space for 10 percent more (about 3), for a total of 29:

```
sp_configure "number of open databases", 29
```

If there is a lot of activity on the server, for example, if databases are being added or dropped, run sp_monitorconfig periodically. You will need to reset the cache size as the number of active descriptors changes.

number of open indexes

Summary Information	
Name in pre-11.0 release	N/A
Default value	500
Range of values	100–2147483647
Status	Dynamic
Display level	Basic
Required role	System Administrator

number of open indexes sets the maximum number of indexes that can be used simultaneously on Adaptive Server.

If you are planning to make a substantial change, such as loading databases with a large number of indexes from another server, you can calculate an estimated metadata cache size by using sp_helpconfig. sp_helpconfig displays the amount of memory required for a given number of metadata descriptors, as well as the number of descriptors that can be accommodated by a given amount of memory. An index metadata descriptor represents the state of an index while it is in use or cached between uses.

Optimizing the *number of open indexes* parameter for your system

The default run value is 500. If this number is insufficient, Adaptive Server displays a message after trying to reuse active index descriptors, and you will need to adjust this value.

In order to configure the number of open indexes parameter optimally, perform the following steps:

1 Use sp_countmetadata to find the total number of index metadata descriptors. For example:

```
sp_countmetadata "open indexes"
```

The best time to run sp_countmetadata is when there is little activity in the server. Running sp_countmetadata during a peak time can cause contention with other processes.

Suppose Adaptive Server reports the following information:

```
There are 698 user indexes in all database(s),
requiring 286.289 Kbytes of memory. The 'open
indexes' configuration parameter is currently set
to 500.
```

2 Configure the number of open indexes parameter to 698 as follows:

```
sp_configure "number of open indexes", 698
```

This new configuration is only a start; the ideal size should be based on the number of *active* index metadata cache descriptors, not the total number of indexes.

3 During a peak period, find the number of active index metadata descriptors. For example:

```
sp_monitorconfig "open indexes"
Usage information at date and time: Jan 14 1997 8:54AM.
Name            # Free   # Active   % Active  # Max Ever Used   Re-used
--------------  --------  ------   --------   ---------------   -------
number of open  182       516      73.92      590               No
indexes
```

In this example, 590 is the maximum number of index descriptors that have been used since the server was last started.

See sp_monitorconfig in the *Adaptive Server Reference Manual* for more information.

4 Configure the number of open indexes configuration parameter to 590, plus additional space for 10 percent more (59), for a total of 649:

```
sp_configure "number of open indexes", 649
```

If there is a lot of activity on the server, for example, if tables are being added or dropped, run sp_monitorconfig periodically. You will need to reset the cache size as the number of active descriptors changes.

number of open objects

Summary Information	
Name in pre-11.0 release	open objects
Default value	500
Range of values	100–2147483647
Status	Dynamic
Display level	Basic
Required role	System Administrator

number of open objects sets the maximum number of objects that can be open simultaneously on Adaptive Server.

If you are planning to make a substantial change, such as loading databases with a large number of objects from another server, you can calculate an estimated metadata cache size by using sp_helpconfig. sp_helpconfig displays the amount of memory required for a given number of metadata descriptors, as well as the number of descriptors that can be accommodated by a given amount of memory. An object metadata descriptor represents the state of an object while it is in use, or cached between uses.

Optimizing the *number of open objects* parameter for your system

The default run value is 500. If this number is insufficient, Adaptive Server displays a message after trying to re-use active object descriptors. You will need to adjust this value.

To set the number of open objects parameter optimally:

1 Use sp_countmetadata to find the total number of object metadata cache descriptors. For example:

```
sp_countmetadata "open objects"
```

The best time to run sp_countmetadata is when there is little activity in the server. Running sp_countmetadata during a peak time can cause contention with other processes.

Suppose Adaptive Server reports the following information:

```
There are 340 user objects in all database(s),
requiring 140.781 Kbytes of memory. The 'open
objects' configuration parameter is currently set
to 500
```

2 Configure the number of open objects parameter to that value, as follows:

```
sp_configure "number of open objects", 357
```

357 covers the 340 user objects, plus 5 percent to accommodate temporary tables.

This new configuration is only a start; the ideal size should be based on the number of *active* object metadata cache descriptors, not the *total* number of objects.

3 During a peak period, find the number of active metadata cache descriptors, for example:

```
sp_monitorconfig "open objects"
Usage information at date and time: Jan 14 1997 8:54AM.
Name            # Free   # Active   % Active  # Max Ever Used   Re-used
--------------  --------  ------   --------   ---------------   -------
```

```
number of open  160       357      71.40      397              No
objects
```

In this example, 397 is the maximum number of object descriptors that have been used since the server was last started.

4 Configure the number of open objects to 397, plus 10 percent (40), for a total of 437:

```
sp_configure "number of open objects", 437
```

If there is a lot of activity on the server, for example, if tables are being added or dropped, run sp_monitorconfig periodically. You will need to reset the cache size as the number of active descriptors changes. See sp_monitorconfig in the *Adaptive Server Reference Manual* for more information.

open index hash spinlock ratio

Summary Information	
Name in pre-11.0 release	N/A
Default value	100
Range of values	1–2147483647
Status	Dynamic
Display level	Basic
Required role	System Administrator

open index hash spinlock ratio sets the number of index metadata descriptor hash tables that are protected by one **spinlock**. This parameter is used for multiprocessing systems only.

All the index descriptors belonging to the table are accessible through a hash table. When a query is run on the table, Adaptive Server uses hash tables to look up the necessary index information in its sysindexes rows. A hash table is an internal mechanism used by Adaptive Server to retrieve information quickly.

Usually, you do not need to change this parameter. In rare instances, however, you may need to reset it if Adaptive Server demonstrates contention from hash spinlocks. You can get information about spinlock contention by using sp_sysmon. For more about sp_sysmon, see the *Performance and Tuning Guide*.

For more information about configuring spinlock ratios, see "Configuring spinlock ratio parameters" on page 644.

open index spinlock ratio

Summary Information	
Name in pre-11.0 release	N/A
Default value	100
Range of values	1–214748364
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

open index spinlock ratio specifies the number of index metadata descriptors that are protected by one **spinlock**.

Adaptive Server uses a spinlock to protect an index descriptor, since more than one process can access the contents of the index descriptor. This parameter is used for multiprocessing systems only.

The value specified for this parameter defines the ratio of index descriptors per spinlock.

If one spinlock is shared by too many index descriptors, it can cause spinlock contention. Use sp_sysmon to get a report on spinlock contention. See the *Performance and Tuning Guide* more information. If sp_sysmon output indicates an index descriptor spinlock contention of more than 3 percent, try decreasing the value of open index spinlock ratio.

For more information about configuring spinlock ratios, see "Configuring spinlock ratio parameters" on page 644.

open object spinlock ratio

Summary Information	
Name in pre-11.0 release	N/A
Default value	100
Range of values	1–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

open object spinlock ratio specifies the number of object descriptors that are protected by one **spinlock**. Adaptive Server uses a spinlock to protect an object descriptor, since more than one process can access the contents of the object descriptor. This configuration parameter is used for multiprocessing systems only.

The default value for this parameter is 100; 1 spinlock for each 100 object descriptors configured for your server. If your server is configured with only one engine, Adaptive Server sets only 1 object descriptor spinlock, regardless of the number of object descriptors.

If one spinlock is shared by too many object descriptors, it causes spinlock contention. Use sp_sysmon to get a report on spinlock contention. See the *Performance and Tuning Guide* for more information on spinlock contention. If sp_sysmon output indicates an object descriptor spinlock contention of more than 3 percent, try decreasing the value of the open object spinlock ratio parameter.

For more information about configuring spinlock ratios, see "Configuring spinlock ratio parameters" on page 644.

Network communication

Use the parameters in this group to configure communication between Adaptive Server and remote servers, and between Adaptive Server and client programs.

allow remote access

Summary Information	
Name in pre-11.0 release	remote access
Default value	1 (on)
Valid values	0 (off), 1 (on)
Status	Dynamic
Display level	Intermediate
Required role	System Security Officer

allow remote access controls logins from remote Adaptive Servers. The default value of 1 allows Adaptive Server to communicate with Backup Server. Only a System Security Officer can set allow remote access.

Setting the value to 0 disables server-to-server RPCs. Since Adaptive Server communicates with Backup Server via RPCs, setting this parameter to 0 makes it impossible to back up a database.

Since other system administration actions are required to enable remote servers other than Backup Server to execute RPCs, leaving this option set to 1 does not constitute a security risk.

allow sendmsg

Summary Information	
Default value	0 (off)
Valid values	0 (off), 1 (on)
Status	Dynamic
Display level	Comprehensive
Required role	System Security Officer

The allow sendmsg parameter enables or disables sending messages from Adaptive Server to a UDP (User Datagram Protocol) port. When allow sendmsg is set to 1, any user can send messages using sp_sendmsg or syb_sendmsg. To set the port number used byAdaptive Server, see "syb_sendmsg port number" on page 157.

Note Sending messages to UDP ports is not supported on Windows NT.

default network packet size

Summary Information	
Name in pre-11.0 release	default network packet size
Default value	512
Range of values	512–524288
Status	Static
Display level	Intermediate
Required role	System Administrator

default network packet size configures the default packet size for all Adaptive Server users. You can set default network packet size to any multiple of 512 bytes; values that are not even multiples of 512 are rounded down.

Memory for all users who log in with the default packet size is allocated from Adaptive Server's memory pool, as set with total logical memory. This memory is allocated for network packets when Adaptive Server is started.

Each Adaptive Server user connection uses:

- One read buffer
- One buffer for messages
- One write buffer

Each of these buffers requires default network packet size bytes. The total amount of memory allocated for network packets is:

```
(number of user connections + number of worker processes) * 3 * default network
packet size
```

For example, if you set the default network packet size to 1024 bytes, and you have 50 user connections and 20 worker processes, the amount of network memory required is:

(50 + 20) * 3 * 1024 = 215040 bytes

If you increase the default network packet size, you must also increase the max network packet size to at least the same size. If the value of max network packet size is greater than the value of default network packet size, to increase the value of additional network memory. See "additional network memory" on page 169 for further information.

Use sp_sysmon to see how changing the default network packet size parameter affects network I/O management and task switching. For example, try increasing default network packet size and then checking sp_sysmon output to see how this affects bcp for large batches. See the *Performance and Tuning Guide* for more information.

Requesting a Larger Packet Size at Login

The default packet size for most client programs like bcp and isql is set to 512 bytes. If you change the default packet size, clients must request the larger packet size when they connect. Use the -A flag to Adaptive Server client programs to request a large packet size. For example:

```
isql -A2048
```

max network packet size

Summary Information	
Name in pre-11.0 release	maximum network packet size
Default value	512
Range of values	512–524288
Status	Static
Display level	Intermediate
Required role	System Administrator

max network packet size specifies the maximum network packet size that can be requested by clients communicating with Adaptive Server.

If some of your applications send or receive large amounts of data across the network, these applications can achieve significant performance improvement by using larger packet sizes. Two examples are large bulk copy operations and applications that read or write large text or image values.

Generally, you want:

- The value of default network packet size to be small for users who perform short queries
- max network packet size to be large enough to allow users who send or receive large volumes of data to request larger packet sizes

max network packet size must always be as large as, or larger than, the default network packet size. Values that are not even multiples of 512 are rounded down.

For client applications that explicitly request a larger network packet size to receive it, you must also configure additional network memory. See "additional network memory" on page 169 for more information.

See bcp and isql in the *Utility Guide* for information on using larger packet sizes from these programs. Open Client Client-Library documentation includes information on using variable packet sizes.

Choosing packet sizes

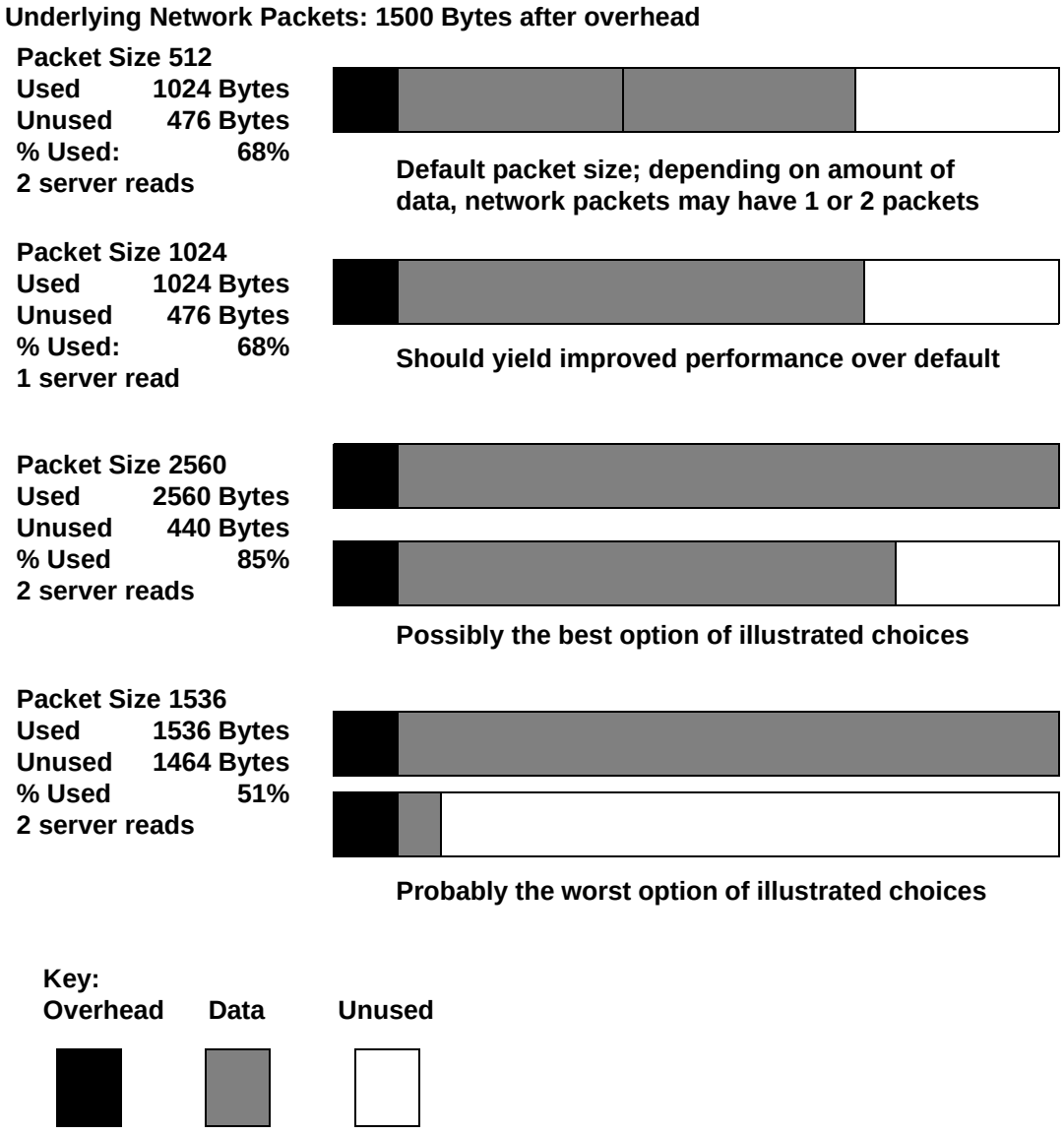
For best performance, choose a server packet size that works efficiently with the underlying packet size on your network. The goals are:

- Reducing the number of server reads and writes to the network

- Reducing unused space in network packets (increasing network throughput)

For example, if your network packet size carries 1500 bytes of data, setting Adaptive Server's packet size to 1024 (512*2) will probably achieve better performance than setting it to 1536 (512*3). Figure 5-5 shows how four different packet size configurations would perform in such a scenario.

Figure 5-5: Factors in determining packet size



After you determine the available data space of the underlying packets on your network, perform your own benchmark tests to determine the optimum size for your configuration.

Use sp_sysmon to see how changing max network packet size affects network I/O management and task switching. For example, try increasing max network packet size and then checking sp_sysmon output to see how this affects bcp for large batches. See the *Performance and Tuning Guide* for more information.

max number network listeners

Summary Information	
Name in pre-11.0 release	cmaxnetworks
Default value	5
Range of values	0–2147483647
Status	Static
Display level	Comprehensive
Required role	System Administrator

max number network listeners specifies the maximum number of network listeners allowed by Adaptive Server at one time.

Each master port has one network listener. Generally, there is no need to have multiple master ports, unless your Adaptive Server needs to communicate over more than one network type. Some platforms support both socket and TLI (Transport Layer Interface) network interfaces. Refer to the configuration documentation for your platform for information on supported network types.

number of remote connections

Summary Information	
Name in pre-11.0 release	remote connections
Default value	20
Range of values	5–32767
Status	Static
Display level	Intermediate
Required role	System Administrator

number of remote connections specifies the number of logical connections that can be open to and from an Adaptive Server at one time. Each simultaneous connection to XP Server for ESP execution uses up to one remote connection each. For more information, see Chapter 13, "Managing Remote Servers."

number of remote logins

Summary Information	
Name in pre-11.0 release	remote logins
Default value	20
Range of values	0–32767
Status	Static
Display level	Intermediate
Required role	System Administrator

number of remote logins controls the number of active user connections from Adaptive Server to remote servers. Each simultaneous connection to XP Server for ESP execution uses up to one remote login each. You should set this parameter to the same (or a lower) value as number of remote connections. For more information, see Chapter 13, "Managing Remote Servers."

number of remote sites

Summary Information	
Name in pre-11.0 release	remote sites
Default value	10
Range of values	0–32767
Status	Static
Display level	Intermediate
Required role	System Administrator

number of remote sites determines the maximum number of remote sites that can access Adaptive Server simultaneously. An each Adaptive Server-to-XP Server connection uses one remote site connection.

Internally, number of remote sites determines the number of site handlers that can be active at any one time; all server accesses from a single site are managed with a single site handler. For more information, see Chapter 13, "Managing Remote Servers."

remote server pre-read packets

Summary Information	
Name in pre-11.0 release	pre-read packets
Default value	3
Range of values	3–32767
Status	Static
Display level	Intermediate
Required role	System Administrator

remote server pre-read packets determines the number of packets that will be "pre-read" by a site handler during connections with remote servers.

All communication between two servers is managed through a single site handler, to reduce the required number of connections. The site handler can pre-read and keep track of data packets for each user process before the receiving process is ready to accept them.

The default value for remote server pre-read packets is appropriate for most servers. Increasing the value uses more memory; decreasing the value can slow network traffic between servers. For more information, see Chapter 13, "Managing Remote Servers."

syb_sendmsg port number

Summary Information	
Default value	0
Valid values	0, or 1024 - 65535, or system limit
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The syb_sendmsg port number parameter specifies the port number that Adaptive Server uses to send messages to a UDP (User Datagram Protocol) port with sp_sendmsg or syb_sendmsg.

If more than one engine is configured, a port is used for each engine, numbered consecutively from the port number specified. If the port number is set to the default value, 0 Adaptive Server assigns port numbers.

Note Sending messages to UDP ports is not supported on Windows NT.

A System Security Officer must set the allow sendmsg configuration parameter to 1 to enable sending messages to UDP ports. To enable UDP messaging, a System Administrator must set allow allow sendmsg to 1. See "allow sendmsg" on page 150. For more information on UDP messaging, see sp_sendmsg in the *Adaptive Server Reference Manual*.

tcp no delay

Summary Information	
Name in pre-11.0 release	T1610 (trace flag)
Default value	0 (off)
Valid values	0 (off), 1 (on)
Status	Static
Display level	Comprehensive
Required role	System Administrator

The tcp no delay parameter controls TCP (Transmission Control Protocol) packet batching. The default value is 0, which means that TCP packets are batched.

TCP normally batches small logical packets into single larger physical packets (by briefly delaying packets) fill physical network frames with as much data as possible. This is intended to improve network throughput in terminal emulation environments where there are mostly keystrokes being sent across the network.

However, applications that use small TDS (Tabular Data Stream™) packets may benefit from disabling TCP packet batching. To disable TCP packet batching, set tcp no delay to 1.

Note Disabling TCP packet batching means that packets will be sent, regardless of size; this will increase the volume of network traffic.

O/S resources

The parameters in this group are related to Adaptive Server's use of operating system resources.

max async i/os per engine

Summary Information	
Name in pre-11.0 release	cnmaxaio_engine
Default value	2147483647
Range of values	1–2147483647
Status	Static
Display level	Comprehensive
Required role	System Administrator

max async i/os per engine specifies the maximum number of outstanding asynchronous disk I/O requests for a single engine at one time. See "max async i/os per server" on page 159 for more information.

max async i/os per server

Summary Information	
Name in pre-11.0 release	cnmaxaio_server
Default value	2147483647
Range of values	1–2147483647
Status	Static
Display level	Comprehensive
Required role	System Administrator

The max async i/os per server parameter specifies the maximum number of asynchronous disk I/O requests that can be outstanding for Adaptive Server at one time. This limit is not affected by the number of online engines per Adaptive Server; max async i/os per server limits the total number of asynchronous I/Os a server can issue at one time, regardless of how many online engines it has. max async i/os per engine limits the number of outstanding I/Os per engine.

Most operating systems limit the number of asynchronous disk I/Os that can be processed at any one time; some operating systems limit the number per operating system process, some limit the number per system, and some do both. If an application exceeds these limits, the operating system returns an error message. Because operating system calls are relatively expensive, it is inefficient for Adaptive Server to attempt to perform asynchronous I/Os that get rejected by the operating system.

To avoid this, Adaptive Server maintains a count of the outstanding asynchronous I/Os per engine and per server; if an engine issues an asynchronous I/O that would exceed either max async i/os per engine or max async i/os per server, Adaptive Server delays the I/O until enough outstanding I/Os have completed to fall below the exceeded limit.

For example, assume an operating system limit of 200 asynchronous I/Os per system and 75 per process and an Adaptive Server with three online engines. The engines currently have a total of 200 asynchronous I/Os pending, distributed according to the following table:

Engine	Number of I/Os pending	Outcome
0	60	Engine 0 delays any further asynchronous I/Os until the total for the server is under the operating system *per-system* limit and then continues issuing asynchronous I/Os.
1	75	Engine 1 delays any further asynchronous I/Os until the per-engine total is under the operating system *per-process* limit and then continues issuing asynchronous I/Os.
2	65	Engine 2 delays any further asynchronous I/Os until the total for server is under the operating system *per-system* limit and then continues issuing asynchronous I/Os.

All I/Os (both asynchronous and synchronous) require a disk I/O structure, so the total number of outstanding disk I/Os is limited by the value of disk i/o structures. It is slightly more efficient for Adaptive Server to delay the I/O because it cannot get a disk I/O structure than because the I/O request exceeds max i/os per server. You should set max async i/os per server equal to the value of disk i/o structures. See "disk i/o structures" on page 110.

If the limits for asynchronous I/O can be tuned on your operating system, make sure they are set high enough for Adaptive Server. There is no penalty for setting them as high as needed.

Use sp_sysmon to see if the per server or per engine limits are delaying I/O on your system. If sp_sysmon shows that Adaptive Server exceeded the limit for outstanding requests per engine or per server, raise the value of the corresponding parameter. See the *Performance and Tuning Guide* for more information.

o/s file descriptors

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	Site-specific
Status	Read-only
Display level	Comprehensive
Required role	System Administrator

o/s file descriptors indicates the maximum per-process number of file descriptors configured for your operating system. This parameter is read-only and cannot be configured through Adaptive Server.

Many operating systems allow you to configure the number of file descriptors available per process. See your operating system documentation for further information on this.

The number of file descriptors available for Adaptive Server connections, which will be less than the value of o/s file descriptors, is stored in the variable *@@max_connections*. For more information on the number of file descriptors available for connections, see "Upper Limit to the maximum number of user connections" on page 223.

shared memory starting address

Summary Information	
Name in pre-11.0 release	mrstart
Default value	0
Range of values	Platform-specific
Status	Static
Display level	Comprehensive
Required role	System Administrator

shared memory starting address determines the virtual address where Adaptive Server starts its shared memory region.

It is unlikely that you will ever have to reconfigure shared memory starting address. You should do so only after consulting with Sybase Technical Support.

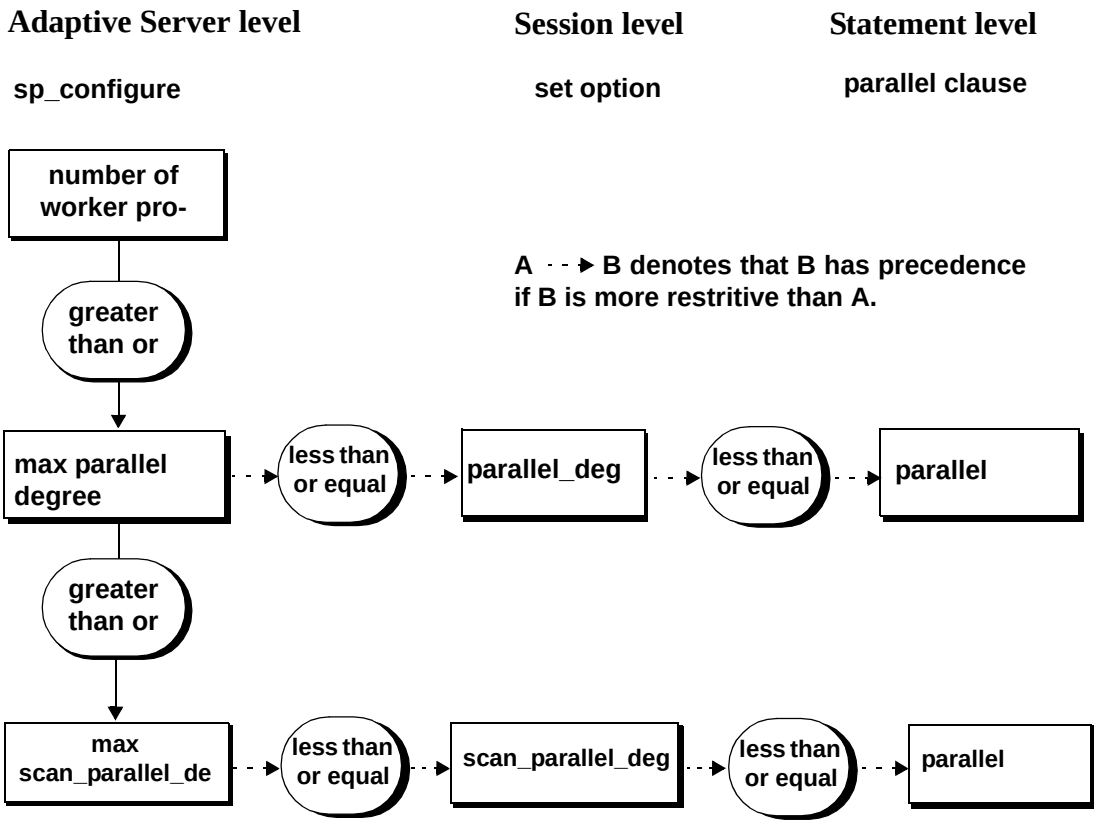
Parallel queries

The following parameters configure Adaptive Server for parallel query processing – where the optimizer evaluates each query to determine whether it is eligible for parallel execution.

To determine the best values to use for the configuration parameters, and to understand how these values affect the optimizer, see Chapter 17, "Adaptive Server Optimizer," and Chapter 23, "Parallel Query Optimization," in the *Performance and Tuning Guide*.

number of worker processes, max parallel degree, and max scan parallel degree control parallel query processing at the server level. Using the parallel_degree, process_limit_action, and scan_parallel_degree options to the set command can limit parallel optimization at the session level, and using the parallel keyword of the select command can limit parallel optimization of specific queries. Figure 5-6 shows the precedence of the configuration parameters and session parameters.

Figure 5-6: Precedence of parallel options



number of worker processes

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0–2147483647
Status	Dynamic
Display level	Basic
Required role	System Administrator

number of worker processes specifies the maximum number of worker processes that Adaptive Server can use at any one time for all simultaneously running parallel queries combined.

Adaptive Server issues a warning message at start-up if there is insufficient memory to create the specified number of worker processes. memory per worker process controls the memory allocated to each worker process.

max parallel degree

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Range of values	1–255
Status	Dynamic
Display level	Basic
Required role	System Administrator

max parallel degree specifies the server-wide maximum number of worker processes allowed per query. This is called the *maximum degree of parallelism*.

If this number is too low, the performance gain for a given query may not be as significant as it could be; if the number is too high, the server may compile plans that require more processes than are actually available at execution time, or the system may become saturated, resulting in decreased throughput. To enable parallel partition scans, set this parameter to be equal to or greater than the number of partitions in the table you are querying.

The value of this parameter must be less than or equal to the current value of number of worker processes.

If you set max parallel degree to 1, Adaptive Server scans all tables or indexes serially.

Changing max parallel degree causes all query plans in the procedure cache to be invalidated, and new plans are compiled the next time you execute a stored procedure or trigger.

For more information on parallel sorting, see Chapter 24, "Parallel Sorting," in the *Performance and Tuning Guide*.

max scan parallel degree

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Range of values	1–255
Status	Dynamic
Display level	Basic
Required role	System Administrator

max scan parallel degree specifies the server-wide maximum degree of parallelism for hash-based scans. Hash-based scans may be used for the following access methods:

- Parallel index scans for partitioned and nonpartitioned tables
- Parallel table scans for nonpartitioned tables

max scan parallel degree applies per table or index; that is, if max scan parallel degree is 3, and one table in a join query is scanned using a hash-based table scan and the second can best be accessed by a hash-based index scan, the query could use 9 worker processes (as long as max scan parallel degree is set to 9 or higher.)

The optimizer uses this parameter as a guideline when it selects the number of processes to use for parallel, nonpartition-based scan operations. It does not apply to parallel sort. Because there is no partitioning to spread the data across devices, parallel processes can be accessing the same device during the scan. This can cause additional disk contention and head movement, which can degrade performance. To prevent multiple disk accesses from becoming a problem, use this parameter to reduce the maximum number of processes that can access the table in parallel.

If this number is too low, the performance gain for a given query will not be as significant as it could be; if the number is too large, the server may compile plans that use enough processes to make disk access less efficient. A general rule of thumb is to set this parameter to no more than 2 or 3, because it takes only 2 to 3 worker processes to fully utilize the I/O of a given physical device.

Set the value of this parameter to less than or equal to the current value of max parallel degree. Adaptive Server returns an error if you specify a number larger than the max parallel degree value.

If you set max scan parallel degree to 1, Adaptive Server does not perform hash-based scans.

Changing max scan parallel degree causes all query plans in the procedure cache to be invalidated, and new plans are compiled the next time you execute a stored procedure or trigger.

memory per worker process

Summary Information	
Name in pre-11.0 release	N/A
Default value	1024
Range of values	1024–2147483647
Status	Dynamic
Display level	Basic
Required role	System Administrator

memory per worker process specifies the amount of memory (in bytes) used by worker processes. Each worker process requires memory for messaging during query processing. This memory is allocated from a shared memory pool; the size of this pool is memory per worker process multiplied by number of worker processes. For most query processing, the default size is more than adequate. If you use dbcc checkstorage, and have set number of worker processes to 1, you may need to increase memory per worker process to 1792 bytes. See "Other configuration parameters for parallel processing" on page 528 of the *Performance and Tuning Guide* for information on setting this parameter.

For more information on Adaptive Server's memory allocation, see Chapter 18, "Configuring Memory."

Physical memory

The parameters in this group configure your machine's physical memory resources.

allocate max shared memory

Summary information	
Name in pre-12.5 release	N/A

Summary information	
Default value	0
Range of values	0,1
Status	Dynamic
Display level	Basic
Required role	System Administrator

allocate max shared memory determines whether Adaptive Server allocates all the memory specified by max memory at start-up or only the amount of memory the configuration parameter requires.

By setting allocate max shared memory to 0, you ensure that Adaptive Server uses only the amount of shared memory required by the current configuration, and allocates only the amount of memory required by the configuration parameters at start-up, which is a smaller value than max memory.

If you set allocate max shared memory to 1, Adaptive Server allocates all the memory specified by max memory at start-up. If allocate max shared memory is 1, and if you increase max memory, Adaptive Server immediately uses additional shared memory segments. This means that Adaptive Server always has the memory required for any memory configuration changes you make and there is no performance degradation while the server readjusts for additional memory. However, if you do not predict memory growth accurately, and max memory is set to a large value, you may waste total physical memory.

dynamic allocation on demand

Summary information	
Name in pre-12.5 release	N/A
Default value	1
Range of values	0, 1
Status	Dynamic
Display level	Basic
Required role	System Administrator

Determines when memory is allocated for changes to dynamic memory configuration parameters.

If you set dynamic allocation on demand to 1, memory is allocated only as it is needed. That is, if you change the configuration for number of user connections from 100 to 200, the memory for each user is added only when the user connects to the server. Adaptive Server continues to add memory until it reaches the new maximum for user connections.

If dynamic allocation on demand is set to 0, all the memory required for any dynamic configuration changes is allocated immediately. That is, when you change the number of user connections from 100 to 200, the memory required for the extra 100 user connections is immediately allocated.

max memory

Summary information	
Name in pre-12.5 release	N/A
Default value	Platform-dependent
Range of values	Platform-dependent minimum – 2147483647
Status	Dynamic
Display level	Basic
Required role	System Administrator

Specifies the maximum amount of total physical memory that you can configure Adaptive Server to allocate. max memory must be greater than the total logical memory consumed by the current configuration of Adaptive Server.

There is no performance penalty for configuring Adaptive Server to use the maximum memory available to it on your computer. However, assess the other memory needs on your system, or Adaptive Server may not be able to acquire enough memory to start.

See Chapter 18, "Configuring Memory," in the *System Administration Guide* for instructions on how to maximize the amount of max memory for Adaptive Server.

If Adaptive Server cannot start

When allocate max shared memory is set to 1, Adaptive Server must have the amount of memory available that is specified by max memory. If the memory is not available, Adaptive Server will not start. If this occurs, reduce the memory requirements for Adaptive Server by manually changing the value of max memory in the server's configuration file. You can also change the value of allocate max shared memory to 0 so that not all memory required by max memory is required at start-up.

You may also want to reduce the values for other configuration parameters that require large amounts of memory. Then restart Adaptive Server to use the memory specified by the new values. If Adaptive Server fails to start because the total of other configuration parameter values is higher than the max memory value, see Chapter 18, "Configuring Memory," in the *System Administration Guide* for information about configuration parameters that use memory.

additional network memory

Summary Information	
Name in pre-11.0 release	additional network memory
Default value	0
Range of values	0–2147483647
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

additional network memory sets the maximum size of additional memory that can be used for network packets that are larger than the default packet size. Adaptive Server rounds down the value you enter to the nearest 2K value. The default value indicates that no extra space is allocated for large packets.

If you increase max network packet size but do not increase additional network memory, clients cannot use packet sizes that are larger than the default size, because all allocated network memory is reserved for users at the default size. Adaptive Server guarantees that every user connection can log in at the default packet size. In this situation, users who request a large packet size when they log in receive a warning message telling them that their application will use the default size.

Increasing additional network memory may improve performance for applications that transfer large amounts of data. To determine the value for additional network memory when your applications use larger packet sizes:

- Estimate the number of simultaneous users who will request the large packet sizes, and the sizes their applications will request,
- Multiply this sum by three, since each connection needs three buffers,
- Add two percent for overhead, and
- Round the value to the next highest multiple of 2048.

For example, if you estimate these simultaneous needs for larger packet sizes:

Application	Packet size	Overhead
bcp	8192	
Client-Library	8192	
Client-Library	4096	
Client-Library	4096	
Total	24576	
Multiply by 3 buffers/user	*3	
	73728	
Compute 2% overhead		* .02=1474
Add overhead	+ 1474	
Additional network memory	75202	
Round up to multiple of 2048	75776	

you should set additional network memory to 75,776 bytes.

heap memory per user

Summary Information	
Name in pre-11.0 release	NA
Default value	4K
Valid values	0 – 2147483647 bytes
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

heap memory per user configures the amount of heap memory per user. A heap memory pool is an internal memory created at startup that tasks use to dynamically allocate memory as needed. This memory pool is important if you are running tasks that use wide columns, which require a lot of memory from the stack. The heap memory allocates a temporary buffer that enables these wide column tasks to finish. The heap memory the task uses is returned to the heap memory pool when the task is finished.

The size of the memory pool depends on the number of user connections. Sybase recommends that you set heap memory per user to three times the size of your logical page.

lock shared memory

Summary Information	
Name in pre-11.0 release	T1611 (trace flag)
Default value	0 (off)
Valid values	0 (off), 1 (on)
Status	Static
Display level	Comprehensive
Required role	System Administrator

lock shared memory disallows swapping of Adaptive Server pages to disk and allows the operating system kernel to avoid the server's internal page locking code. This can reduce disk reads, which are expensive.

Not all platforms support shared memory locking. Even if your platform does, lock shared memory may fail due to incorrectly set permissions, insufficient physical memory, or for other reasons. See the configuration documentation for your platform for information on shared memory locking.

max SQL text monitored

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0–2147483647
Status	Static
Display level	Comprehensive

Summary Information	
Required role	System Administrator

max SQL text monitored specifies the amount of memory allocated per user connection for saving SQL text to memory shared by Adaptive Server Monitor.

Initially, the amount of memory allocated for saving text is 0, and since this parameter is static, you must restart Adaptive Server before you can start saving SQL Text.

If you do not allocate enough memory for the batch statements, the text you want to view may be in the section of the batch that is truncated. Sybase recommends an initial value of 1024 bytes of memory per user connection.

The total memory allocated from shared memory for the SQL text is the product of max SQL text monitored multiplied by the currently configured number of user connections.

For more information on max SQL text monitored, see "Configuring Adaptive Server to save SQL batch text" on page 64.

total physical memory

Summary information	
Name in pre-12.5 release	total memory
Default value	N/A
Range of values	N/A
Status	Read-only
Display level	Intermediate
Required role	System Administrator

total physical memory is a read-only configuration parameter that displays the total physical memory for the current configuration of Adaptive Server. The total physical memory is the amount of memory that Adaptive Server is using at a given moment in time. Adaptive Server should be configured so that the value for max memory is larger than the value for total logical memory, and the value for total logical memory is larger than the value for total physical memory.

total logical memory

Summary information	
Name in pre-12.5 release	total memory
Default value	N/A
Range of values	N/A
Status	Read-only
Display level	Intermediate
Required role	System Administrator

total logical memory displays the total logical memory for the current configuration of Adaptive Server. The total logical memory is the amount of memory that Adaptive Server's current configuration uses. total logical memory displays the memory which is required to be available, but which may or may not be in use at any given moment. For information about the amount of memory in use at a given moment, see the configuration parameter total physical memory. You cannot use total logical memory to set any of the memory configuration parameters.

Processors

The parameters in this group configure processors in an SMP environment.

number of engines at startup

Summary information	
Name in pre-12.5 release	N/A
Default value	1
Range of values	1 – number of CPUs on machine
Status	Static
Display level	Basic
Required role	System Administrator

Adaptive Server allows users to take all engines offline, except engine zero.

number of engines at startup is used exclusively during start-up to set the number of engines brought online. It is designed to allow users the greatest flexibility in the number of engines brought online, subject to the restriction that you cannot set the value of number of engines at startup to a value greater than the number of CPUs on your machine, or to a value greater than the configuration of max online engines. Users who do not intend to bring engines online after start-up should set max online engines and number of engines at startup to the same value. A difference between number of engines at startup and max online engines wastes approximately 1.8 MB of memory per engine.

max online engines

Summary Information	
Name in pre-11.0 release	max online engines
Default value	1
Range of values	1–128
Status	Static
Display level	Intermediate
Required role	System Administrator

The role of max online engines is to set a high value of engines to be taken online at any one time in an SMP environment. It does not take the number of CPUs available at start-up into account, and allows users to add CPUs at a later date.

max engines online specifies the maximum number of Adaptive Server engines that can be online at any one time in an SMP environment. See Chapter 20, "Managing Multiprocessor Servers," for a detailed discussion of how to set this parameter for your SMP environment.

At start-up, Adaptive Server starts with a single engine and completes its initialization, including recovery of all databases. Its final task is to allocate additional server engines. Each engine accesses common data structures in shared memory.

When tuning the max engines online parameter:

- Never have more online engines than there are CPUs.
- Depending on overall system load (including applications other than Adaptive Server), you may achieve optimal throughput by leaving some CPUs free to run non-Adaptive Server processes.

- Better throughput can be achieved by running fewer engines with high CPU use, rather than by running more engines with low CPU use.
- Scalability is application-dependent. You should conduct extensive benchmarks on your application to determine the best configuration of online engines.
- You can use the dbcc engine command to take engines offline or to bring them online line. You can offline all engines other than engine zero.

See "Taking engines offline with dbcc engine" on page 639 for information on using dbcc engine. See Chapter 3, "Using Engines and CPUs," in the *Performance and Tuning Guide* for more information on performance and engine tuning.

RepAgent thread administration

The parameter in this group configures replication via Replication Server®.

enable rep agent threads

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0–1
Status	Dynamic
Display level	Basic
Required role	System Administrator

enable rep agent threads enables the RepAgent thread within Adaptive Server.

Through version 11.0.3 of Replication Server, the Log Transfer Manager (LTM), a replication system component, transfers replication data to the Replication Server. Beginning with Replication Server versions later than 11.0.3, transfer of replication data handled by RepAgent, which will run as a thread under Adaptive Server. Setting enable rep agent threads enables this feature.

Other steps are also required to enable replication. For more information, see the Replication Server documentation.

SQL server administration

The parameters in this group are related to general Adaptive Server administration.

abstract plan cache

Summary Information	
Default value	0
Range of values	0–1
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

abstract plan cache enables caching of abstract plan hash keys. By default, caching is not enabled. For more information, see Chapter 30, "Creating and Using Abstract Plans," in the *Performance and Tuning Guide*. abstract plan load must be enabled in order for plan caching to take affect.

abstract plan dump

Summary Information	
Default value	0
Range of values	0–1
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

abstract plan dump enables the saving of abstract plans to the ap_stdout abstract plans group. For more information, see Chapter 30, "Creating and Using Abstract Plans," in the *Performance and Tuning Guide*.

abstract plan load

Summary Information	
Default value	0
Range of values	0–1
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

abstract plan load enables association of queries with abstract plans in the ap_stdin abstract plans group. For more information, see Chapter 30, "Creating and Using Abstract Plans," in the *Performance and Tuning Guide*.

abstract plan replace

Summary Information	
Default value	0
Range of values	0–1
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

abstract plan replace enables plan replacement for abstract plans in the ap_stdout abstract plans group. For more information, see Chapter 30, "Creating and Using Abstract Plans," in the *Performance and Tuning Guide*. abstract plan load must be enabled in order for replace mode to take effect.

allow backward scans

Summary Information	
Name in pre-11.0 release	N/A
Default value	1 (on)
Valid values	0 (off), 1 (on)
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

allow backward scans controls how the optimizer performs select queries that contain the order by...desc command:

- When the value is set to 1, the optimizer can access the index or table rows by following the page chain in descending index order.
- When the value is set to 0, the optimizer selects the rows into a worktable by following the index page pointers in ascending order and then sorts the worktable in descending order.

The first method—performing backward scans—can speed access to tables that need results ordered by descending column values. Some applications, however, may experience deadlocks due to backward scans. In particular, look for increased deadlocking if you have delete or update queries that scan forward using the same index. There may also be deadlocks due to page splits in the index.

You can use print deadlock information to send messages about deadlocks to the error log. See "print deadlock information" on page 204. Alternatively, you can use sp_sysmon to check for deadlocking. See the *Performance and Tuning Guide* for more information on deadlocks.

allow nested triggers

Summary Information	
Name in pre-11.0 release	nested trigger
Default value	1 (on)
Valid values	0 (off), 1 (on)
Status	Static
Display level	Intermediate
Required role	System Administrator

allow nested triggers controls the use of nested triggers. When the value is set to 1, data modifications made by triggers can fire other triggers. Set allow nested triggers to 0 to disable nested triggers. A set option, self_recursion, controls whether the modifications made by a trigger can cause that trigger to fire again.

allow resource limits

Summary Information	
Name in pre-11.0 release	N/A

Summary Information	
Default value	0 (off)
Valid values	0 (off), 1 (on)
Status	Static
Display level	Comprehensive
Required role	System Administrator

allow resource limits controls the use of resource limits. When the value is set to 1, the server allocates internal memory for time ranges, resource limits, and internal server alarms. The server also internally assigns applicable ranges and limits to user sessions. The output of sp_configure displays the optimizer's cost estimate for a query. Set allow resource limits to 0 to disable resource limits.

allow updates to system tables

Summary Information	
Name in pre-11.0 release	allow updates
Default value	0 (off)
Valid values	0 (off), 1 (on)
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

allow updates to system tables enables users with the System Administrator role to make changes to the system tables and to create stored procedures that can modify system tables. A database administrator can update system tables in any tables that he or she owns if allow updates to system tables is enabled.

System tables include:

- All Sybase-supplied tables in the master database

- All tables in user databases that begin with "sys" and that have an ID value in the sysobjects table of less than or equal to 100

Warning! Incorrect alteration of a system table can result in database corruption and loss of data. Always use begin transaction when modifying a system table to protect against errors that could corrupt your databases. Immediately after finishing your modifications, disable allow updates to system tables.

Stored procedures and triggers you create while allow updates to system tables is set on are always able to update the system tables, even after the parameter has been set off. When you set allow updates to system tables to on, you create a "window of vulnerability," a period of time during which users can alter system tables or create a stored procedure with which the system tables can be altered in the future.

Because the system tables are so critical, it is best to set this parameter to on only in highly controlled situations. To guarantee that no other users can access Adaptive Server while the system tables can be directly updated, restart Adaptive Server in single-user mode. For details, see startserver and dataserver in the *Utility Guide*.

cpu accounting flush interval

Summary Information	
Name in pre-11.0 release	cpu flush
Default value	200
Range of values	1–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

cpu accounting flush interval specifies the amount of time, in *machine* clock ticks, that Adaptive Server waits before flushing CPU usage statistics for each user from sysprocesses to syslogins, a procedure used in chargeback accounting. (Note that this is measured in *machine* clock ticks, not Adaptive Server clock ticks.)

When a user logs in to Adaptive Server, the server begins accumulating figures for CPU usage for that user process in sysprocesses. When a user logs off Adaptive Server, or when the value of cpu accounting flush interval is exceeded, the accumulated CPU usage statistics are flushed from sysprocesses to syslogins. These statistics continue accumulating in syslogins until you clear the totals using sp_clearstats. You can display the current totals from syslogins using sp_reportstats.

The value to which you set cpu accounting flush interval depends on the type of reporting you intend to do. If you intend to run reports on a monthly basis, set cpu accounting flush interval to a relatively high value. With infrequent reporting, it is less critical that the data in syslogins be updated as often.

On the other hand, if you intend to do periodic ad hoc selects on the totcpu column in syslogins to determine CPU usage by process, set cpu accounting flush interval to a lower value. Doing so increases the likelihood of the data in syslogins being up to date when you execute your selects.

Setting cpu accounting flush interval to a low value may cause processes to be mistakenly identified as potential deadlock victims by the lock manager. When the lock manager detects a deadlock, it checks the amount of CPU time accumulated by each competing processes. The process with the lesser amount is chosen as the deadlock victim and is terminated by the lock manager. Additionally, when cpu accounting flush interval is set to a low value, the task handlers that store CPU usage information for processes are initialized more frequently, thus making processes appear as if they have accumulated less CPU time than they actually have. Because of this, the lock manager may select a process as the deadlock victim when, in fact, that process has more accumulated CPU time than the competing process.

If you do not intend to report on CPU usage at all, set cpu accounting flush interval to its maximum value. This reduces the number of times syslogins is updated and reduces the number of times its pages need to be written to disk.

cpu grace time

Summary Information	
Name in pre-11.0 release	ctimemax
Default value	500
Range of values	0–2147483647

Summary Information	
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

cpu grace time, together with time slice, specifies the maximum amount of time that a user process can run without yielding the CPU before Adaptive Server preempts it and terminates it with a time-slice error. The units for cpu grace time are time ticks, as defined by sql server clock tick length. See "sql server clock tick length" on page 207 for more information.

When a process exceeds cpu grace time Adaptive Server "infects" it by removing the process from the internal queues. The process is killed, but Adaptive Server is not affected. This prevents runaway processes from monopolizing the CPU. If any of your user processes become infected, you may be able to temporarily fix the problem by increasing the value of cpu grace time. However, you must be sure that the problem really is a process that takes more than the current value of cpu grace time to complete, rather than a runaway process.

Temporarily increasing the cpu grace time value is a workaround, not a permanent fix, since it may cause other complications; see "time slice" on page 208. Also, see Chapter 3, "Using Engines and CPUs," and "Execution task scheduling" on page 29 of the *Performance and Tuning Guide* for a more detailed discussion of task scheduling.

default database size

Summary Information	
Name in pre-11.0 release	database size
Default value	Logical page size
Range of values	2^a –10000 a. Minimum determined by server's logical page size.
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

default database size sets the default number of megabytes allocated to a new user database if the create database statement is issued without any size parameters. A database size given in a create database statement takes precedence over the value set by this configuration parameter.

If most of the new databases on your Adaptive Server require more than one logical page size, you may want to increase the default.

Note If you alter the model database, you must also increase the default database size, because the create database command copies model to create a new user database.

default fill factor percent

Summary Information	
Name in pre-11.0 release	fillfactor
Default value	0
Range of values	0–100
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

default fill factor percent determines how full Adaptive Server makes each index page when it is creating a new index on existing data, unless the fill factor is specified in the create index statement. The fillfactor percentage is relevant only at the time the index is created. As the data changes, the pages are not maintained at any particular level of fullness.

default fill factor percent affects:

- The amount of storage space used by your data – Adaptive Server redistributes the data as it creates the clustered index.
- Performance – splitting up pages uses Adaptive Server resources.

There is seldom a reason to change default fill factor percent, especially since you can override it in the create index command. For more information about the fill factor percentage, see create index in the *Adaptive Server Reference Manual*.

default exp_row_size percent

Summary Information	
Default value	5
Range of values	0–100

Summary Information	
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

default exp_row_size percent reserves space for expanding updates in data-only-locked tables, to reduce row forwarding. An *expanding update* is any update to a data row that increases the length of the row. Data rows that allow null values or that have variable-length columns may be subject to expanding updates. In data-only-locked tables, expanding updates can require row forwarding if the data row increases in size so that it no longer fits on the page.

The default valu , sets aside 5 percent of the available data page size for use by expanding updates. Since 2002 bytes are available for data storage on pages in data-only-locked tables, this leaves 100 bytes for expansion. This value is only applied to pages for tables that have variable-length columns.

Valid values are 0–99. Setting default exp_row_size percent to 0 means that all pages are completely filled and no space is left for expanding updates.

default exp_row_size percent is applied to data-only-locked tables with variable-length columns when exp_row_size is not explicitly provided with create table or set with sp_chgattribute. If a value is provided with create table, that value takes precedence over the configuration parameter setting. See the Performance and Tuning Guide for more information.

dump on conditions

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0–1
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

dump on conditions determines whether Adaptive Server generates a dump of data in shared memory when it encounters the conditions specified in maximum dump conditions.

Note The dump on conditions parameter is included for use by Sybase Technical Support only. Do not modify it unless you are instructed to do so by Sybase Technical Support.

enable sort-merge joins and JTC

Summary Information	
Default value	0
Range of values	0–1
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

enable sort-merge joins and JTC configuration parameter determines whether merge joins and join transitive closure are considered by the query optimizer. By default, merge joins and join transitive closure are not enabled. To enable merge joins, set this parameter to 1.

Merge joins and join transitive closure can improve performance for queries that access large amounts of data, but increase optimization time. The session-level options set sort-merge on and set jtc on take precedence over the server-wide setting. For more information, see "Enabling and disabling merge joins" on page 423 and "Enabling and disabling join transitive closure" on page 424 in the *Performance and Tuning Guide*.

enable row level access control

Summary Information	
Name in pre-12.5 release	N/A
Default value	0
Valid values	0 (off), 1 (on)
Status	Dynamic
Display level	Comprehensive
Required role	System Security Officer

Enables row level access control. You must have the security services license key enabled before you configure enable row level access control.

enable ssl

Summary Information	
Name in pre-12.5 release	N/A
Default value	0
Valid values	0 (off), 1 (on)
Status	Static
Display level	Comprehensive
Required role	System Security Officer

The enable ssl parameter enables or disables Secure Sockets Layer session-based security.

event buffers per engine

Summary Information	
Name in pre-11.0 release	N/A
Default value	100
Range of values	1–2147483647
Status	Static
Display level	Comprehensive
Required role	System Administrator

The event buffers per engine parameter specifies the number of events per Adaptive Server engine that can be monitored simultaneously by Adaptive Server Monitor. Events are used by Adaptive Server Monitor for observing Adaptive Server performance; if you are not using Adaptive Server Monitor, set this parameter to 1.

The value to which you set event buffers per engine depends on the number of engines in your configuration, the level of activity on your Adaptive Server, and the kinds of applications you are running.

Setting event buffers per engine to a low value may result in the loss of event information. The default value, is likely to be too low for most sites. Values of 2000 and above may be more reasonable for general monitoring. However, you need to experiment to determine the appropriate value for your site.

In general, setting event buffers per engine to a high value may reduce the amount of performance degradation that Adaptive Server Monitor causes Adaptive Server.

Each event buffer uses 100 bytes of memory. To determine the total amount of memory used by a particular value for event buffers per engine, multiply the value by the number of Adaptive Server engines in your configuration.

housekeeper free write percent

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Range of values	0–100
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

housekeeper free write percent specifies the maximum percentage by which the housekeeper task can increase database writes.

For example, to stop the housekeeper task from working when the frequency of database writes reaches 5 percent above normal, set housekeeper free write percent to 5:

```
sp_configure "housekeeper free write percent", 5
```

When Adaptive Server has no user tasks to process, the housekeeper task automatically begins writing changed pages from cache to disk. These writes result in improved CPU utilization, decreased need for buffer washing during transaction processing, and shorter checkpoints.

In applications that repeatedly update the same database page, the housekeeper may initiate some unnecessary database writes. Although these writes occur only during the server's idle cycles, they may be unacceptable on systems with overloaded disks.

The table and index statistics that are used to optimize queries are maintained in memory structures during query processing. When these statistics change, the changes are not written to the systabstats table immediately, to reduce I/O contention and improve performance. Instead, the housekeeper task periodically flushes statistics to disk.

Warning! Setting housekeeper free write percent to 0 disables flushing statistics to the systabstats table. This can seriously impair performance if statistics change significantly.

The default value allows the housekeeper task to increase disk I/O by a maximum of 1 percent. This results in improved performance and recovery speed on most systems.

To disable the housekeeper task, set the value of housekeeper free write percent to 0:

```
sp_configure "housekeeper free write percent", 0
```

You should set this value to 0 only if disk contention on your system is high, and it cannot tolerate the extra I/O generated by the housekeeper.

If you disable the housekeeper tasks, be certain that statistics are kept current. Commands that write statistics to disk are:

- update statistics
- dbcc checkdb (for all tables in a database) or dbcc checktable (for a single table)
- sp_flushstats

You should run one of these commands on any tables that have been updated since the last time statistics were written to disk, at the following times:

- Before dumping a database
- Before an orderly shutdown
- After rebooting, following a failure or orderly shutdown; in these cases, you cannot use sp_flushstats, you must use update statistics or dbcc commands
- After any significant changes to a table, such as a large bulk copy operation, altering the locking scheme, deleting or inserting large numbers of rows, or a truncate table command

To allow the housekeeper task to work continuously, regardless of the percentage of additional database writes, set housekeeper free write percent to 100:

```
sp_configure "housekeeper free write percent", 100
```

Use sp_sysmon to monitor housekeeper performance. See the *Performance and Tuning Guide* for more information.

It might also be helpful to look at the number of free checkpoints initiated by the housekeeper task. The *Performance and Tuning Guide* describes this output.

enable HA

Summary Information	
Default value	0
Range of values	0–1
Status	Static
Display level	Comprehensive
Required role	System Administrator

Setting enable HA is set to 1 allows you to configure Adaptive Server as a companion server in a high availability subsystem. Adaptive Server use s Sybase's Failover to interact with the high availability subsystem. You must set enable HA to 1 before you run the *installhasvss* script (*insthasv* on Windows NT), which installs the system procedures for Sybase's Failover.

Note The license information and the Run value for enable HA are independent of each other. Whether or not you have a license for Sybase Failover, the Run value and the Config value are set to 1 after you reboot Adaptive Server. And until you have a license, you cannot run Sybase Failover. If you have not installed a valid license, Adaptive Server logs an error message and does not activate the feature. See your Installation Guide for information about installing license keys.

Note that, setting enable HA to 1 does not mean that Adaptive Server is configured to work in a high availabiltiy system. You must perform the steps described in *Using Sybase Failover in A High Availability System* to configure Adaptive Server to be a companion server in a high availability system.

When enable HA is set to 0, you cannot configure for Sybase's Failover, and you cannot run *installhasvss* (*insthasv* on Windows NT).

enable housekeeper GC

Summary Information	
Default value	1
Range of values	0–1
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

When enable housekeeper GC is set to 1, the housekeeper task performs space reclamation on data-only-locked tables. housekeeper free write percent must also be set to greater than 0; if it is set to zero, the housekeeper task is disabled. When a user task deletes a row from a data-only-locked table, a task is queued to the housekeeper to check the data and index pages for committed deletes.

When enable housekeeper GC is set to 0, the housekeeper does not perform space reclamation. If all tables on your server use the allpages locking scheme, or if very few deletes or shrinking updates are performed on data-only-locked tables, setting enable housekeeper GC to 0 improves performance by slightly reducing housekeeper overhead. Use this setting:

- If you use only allpages locking
- If there are few deletes performed on your data-only-locked tables
- If your workload leaves little idle CPU time

sp_sysmon reports on how often the housekeeper task performed space reclamation and how many pages were reclaimed. See Performance and Tuning Guide.

identity burning set factor

Summary Information	
Name in pre-11.0 release	identity burning set factor
Default value	5000
Range of values	1–9999999
Status	Static

Summary Information	
Display level	Intermediate
Required role	System Administrator

IDENTITY columns are of type numeric and scale zero whose values are generated by Adaptive Server. Column values can range from a low of 1 to a high determined by the column precision.

For each table with an IDENTITY column, Adaptive Server divides the set of possible column values into blocks of consecutive numbers, and makes one block at a time available in memory. Each time you insert a row into a table, Adaptive Server assigns the IDENTITY column the next available value from the block. When all the numbers in a block have been used, the next block becomes available.

This method of choosing IDENTITY column values improves server performance. When Adaptive Server assigns a new column value, it reads the current maximum value from memory and adds 1. Disk access becomes necessary only after all values within the block have been used. Because all remaining numbers in a block are discarded in the event of server failure (or shutdown with nowait), this method can lead to gaps in IDENTITY column values.

Use identity burning set factor to change the percentage of potential column values that is made available in each block. This number should be high enough for good performance, but not so high that gaps in column values are unacceptably large. The default value, 5000, releases .05 percent of the potential IDENTITY column values for use at one time.

To get the correct value for sp_configure, express the percentage in decimal form, and then multiply it by 10^7 (10,000,000). For example, to release 15 percent (.15) of the potential IDENTITY column values at a time, specify a value of .15 times 10^7 (or 1,500,000) in sp_configure:

```
sp_configure "identity burning set factor", 1500000
```

identity grab size

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Range of values	1–2147483647
Status	Dynamic

Summary Information	
Display level	Intermediate
Required role	System Administrator

identity grab size allows each Adaptive Server process to reserve a block of IDENTITY column values for inserts into tables that have an IDENTITY column.

This is useful if you are doing inserts, and you want all the inserted data to have contiguous IDENTITY numbers. For instance, if you are entering payroll data, and you want all records associated with a particular department to be located within the same block of rows, set identity grab size to the number of records for that department.

identity grab size applies to all users on Adaptive Server. Large identity grab size values result in large gaps in the IDENTITY column when many users insert data into tables with IDENTITY columns.

Sybase recommends setting identity grab size to a value large enough to accommodate the largest group of records you want to insert into contiguous rows.

i/o accounting flush interval

Summary Information	
Name in pre-11.0 release	i/o flush
Default value	1000
Range of values	1–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

i/o accounting flush interval specifies the amount of time, in *machine* clock ticks, that Adaptive Server waits before flushing I/O statistics for each user from sysprocesses to syslogins. This is used for chargeback accounting.

When a user logs in to Adaptive Server, the server begins accumulating I/O statistics for that user process in sysprocesses. When the value of i/o accounting statistics interval is exceeded, or a user logs off Adaptive Server, the accumulated I/O statistics for that user are flushed from sysprocesses to syslogins. These statistics continue accumulating in syslogins until you clear the totals by using sp_clearstats. You display the current totals from syslogins by using sp_reportstats.

The value to which you set i/o accounting flush interval depends on the type of reporting you intend to do. If you intend to run reports on a monthly basis, i/o accounting flush interval to a relatively high value. This is because, with infrequent reporting, it is less critical that the data in syslogins be updated frequently.

If you intend to do periodic ad hoc selects on the totio column syslogins to determine I/O volume by process, to set i/o accounting flush interval to a lower value. Doing so increases the likelihood of the data in syslogins being up to date when you execute your selects.

If you do not intend to report on I/O statistics at all, set i/o accounting flush interval to its maximum value. This reduces the number of times syslogins is updated and the number of times its pages need to be written to disk.

i/o polling process count

Summary Information	
Name in pre-11.0 release	cmaxscheds
Default value	10
Range of values	1–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

i/o polling process count specifies the maximum number of processes that can be run by Adaptive Server before the scheduler checks for disk and/or network I/O completions. Tuning i/o polling process count affects both the response time and throughput of Adaptive Server.

Adaptive Server checks for disk or network I/O completions:

- If the number of tasks run since the last time Adaptive Server checked for I/O completions equals the value for i/o polling process count, and
- At every Adaptive Server clock tick.

As a general rule, increasing the value of i/o polling process count may increase throughput for applications that generate a lot of disk and network I/O. Conversely, decreasing the value may improve process response time in these applications, possibly at the risk of lowering throughput.

If your applications create both I/O and CPU-bound tasks, tuning i/o polling process count to a low value (1–2) ensures that I/O-bound tasks get access to CPU cycles.

For OLTP applications (or any I/O-bound application with user connections and short transactions), tuning i/o polling process count to a value in the range of 20–30 may increase throughput, but it may also increase response time.

When tuning i/o polling process count, consider three other parameters:

- sql server clock tick length, which specifies the duration of Adaptive Server's clock tick in microseconds. See "sql server clock tick length" on page 207.
- time slice, which specifies the number of clock ticks Adaptive Server's scheduler allows a user process to run. See "time slice" on page 208.
- cpu grace time, which specifies the maximum amount of time (in clock ticks) a user process can run without yielding the CPU before Adaptive Server preempts it and terminates it with a time-slice error. See "cpu grace time" on page 181.

Use sp_sysmon to determine the effect of changing the i/o polling process count parameter. See the *Performance and Tuning Guide* for more information.

page lock promotion HWM

Summary Information	
Default value	200
Range of values	2–2147483647
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

page lock promotion HWM (high-water mark), together with the page lock promotion LWM (low-water mark) and page lock promotion PCT (percentage), specifies the number of page locks permitted during a single scan session of a page-locked table or index before Adaptive Server attempts to escalate from page locks to a table lock.

page lock promotion HWM sets a maximum number of page locks allowed on a table before Adaptive Server attempts to escalate to a table lock. When the number of page locks acquired during a scan session exceeds page lock promotion HWM, Adaptive Server attempts to acquire a table lock. The page lock promotion HWM value cannot be higher than number of locks value.

For more detailed information on scan sessions and setting up page lock promotion limits, see "Configuring locks and lock promotion thresholds" on page 226 in the *Performance and Tuning Guide*.

The default value for page lock promotion HWM is appropriate for most applications. You might want to raise the value to avoid table locking. For example, if you know that there are regular updates to 500 pages of an allpages-locked or datapages-locked table containing thousands of pages, you can increase concurrency for the tables by setting page lock promotion HWM to 500 so that lock promotion does not occur at the default setting of 200.

You can also configure lock promotion of page-locked tables and views at the per-object level. See sp_setrowlockpromote in the Adaptive Server Reference Manual.

Use sp_sysmon to see how changing page lock promotion HWM affects the number of lock promotions. sp_sysmon reports the ratio of exclusive page to exclusive table lock promotions and the ratio of shared page to shared table lock promotions. See "Lock promotions" on page 972 in the *Performance and Tuning Guide*.

page lock promotion LWM

Summary Information	
Default value	200
Range of values	2–value of page lock promotion HWM
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

The page lock promotion LWM low-water mark) parameter, together with the page lock promotion HWM (high-water mark) and the page lock promotion PCT, specify the number of page locks permitted during a single scan session of a page locked table or an index before Adaptive Server attempts to promote from page locks to a table lock.

The page lock promotion LWM sets the number of page locks below which Adaptive Server does not attempt to issue a table lock on an object. The page lock promotion LWM must be less than or equal to page lock promotion HWM.

For more information on scan sessions and setting up lock promotion limits, see "Configuring locks and lock promotion thresholds" on page 226 in the *Performance and Tuning Guide*.

The default value for page lock promotion LWM is sufficient for most applications. If Adaptive Server runs out of locks (except for an isolated incident), you should increase number of locks. See the Performance and Tuning Guide for more information.

You can also configure page lock promotion at the per-object level. See sp_setpglockpromote in the *Adaptive Server Reference Manual*.

page lock promotion PCT

Summary Information	
Default value	100
Range of values	1–100
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

If the number of locks held on an object is between page lock promotion LWM (low-water mark) and page lock promotion HWM (high-water mark). page lock promotion PCT sets the percentage of page locks (based on the table size) above which Adaptive Server attempts to acquire a table lock.

For more detailed information on setting up page lock promotion limits, see"Configuring locks and lock promotion thresholds" on page 226 in the *Performance and Tuning Guide*.

The default value for page lock promotion PCT is appropriate for most applications.

You can also configure lock promotion at the per-object level for page locked objects. See sp_setpglockpromote in the *Adaptive Server Reference Manual*.

maximum dump conditions

Summary Information	
Name in pre-11.0 release	N/A
Default value	10
Range of values	10–100
Status	Static
Display level	Intermediate
Required role	System Administrator

The maximum dump conditions parameter sets the maximum number of conditions you can specify under which Adaptive Server generates a dump of data in shared memory.

Note This parameter is included for use by Sybase Technical Support only. Do not modify it unless you are instructed to do so by Sybase Technical Support.

number of alarms

Summary Information	
Name in pre-11.0 release	cnalarm
Default value	40
Range of values	40–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

number of alarms specifies the number of alarm structures allocated by Adaptive Server.

The Transact-SQL command waitfor defines a specific time, time interval, or event for the execution of a statement block, stored procedure, or transaction. Adaptive Server uses alarms to execute waitfor commands correctly. Other internal processes require alarms.

When Adaptive Server needs more alarms than are currently allocated, this message is written to the error log:

```
uasetalarm: no more alarms available
```

The number of bytes of memory required for each is small. If you raise the number of alarms value significantly, you should adjust max memory accordingly.

number of aux scan descriptors

Summary Information	
Default value	200
Range of values	0–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

number of aux scan descriptors sets the number of auxiliary scan descriptors available in a pool shared by all users on a server.

Each user connection and each worker process has 48 scan descriptors exclusively allocated to it. Of these, 16 are reserved for user tables, 12 are reserved for worktables, and 20 are reserved for system tables (with 4 of these set aside for rollback conditions). A descriptor is needed for each table referenced, directly or indirectly, by a query. For user tables, a table reference includes the following:

- All tables referenced in the from clause of the query
- All tables referenced in a view named in the query (the view itself is not counted)
- All tables referenced in a subquery
- All tables that need to be checked for referential integrity (these are used only for inserts, updates, and deletes)
- A table created with select...into
- All worktables created for the query

If a table is referenced more than once (for example, in a self-join, in more than one view, or in more than one subquery) the table is counted each time. If the query includes a union, each select statement in the union query is a separate scan. If a query runs in parallel, the coordinating process and each worker process needs a scan descriptor for each table reference.

When the number of user tables referenced by a query scan exceeds 16, or the number of worktables exceeds 12, scan descriptors from the shared pool are allocated. Data-only-locked tables also require a system table descriptor for each data-only-locked table accessed via a table scan (but not those accessed via an index scan). If more than 16 data-only-locked tables are scanned using table scans in a query, auxiliary scan descriptors are allocated for them.

If a scan needs auxiliary scan descriptors after it has used its allotted number, and there are no descriptors available in the shared pool, Adaptive Server displays an error message and rolls back the user transaction.

If none of your queries need additional scan descriptors, you may still want to leave number of aux scan descriptors set to the default value in case your system requirements grow. Set it to 0 only if you are sure that users on your system will not be running queries on more than 16 tables and that your tables have few or no referential integrity constraints. See "Monitoring scan descriptor usage" on page 199 for more information.

If your queries need more scan descriptors, use one of the following methods to remedy the problem:

- Rewrite the query, or break it into steps using temporary tables. For data-only-locked tables, consider adding indexes if there are many table scans.
- Redesign the table's schema so that it uses fewer scan descriptors, if it uses a large number of referential integrity constraints. You can find how many scan descriptors a query would use by enabling set showplan, noexec on before running the query.
- Increase the number of aux scan descriptors setting.

The following sections describe how to monitor the current and high-water-mark usage with sp_monitorconfig to avoid running out of descriptors and how to estimate the number of scan descriptors you need.

Monitoring scan descriptor usage

sp_monitorconfig reports the number of unused (free) scan descriptors, the number of auxiliary scan descriptors currently being used, the percentage that is active, and the maximum number of scan descriptors used since the server was last started. Run it periodically, at peak periods, to monitor scan descriptor use.

This example output shows scan descriptor use with 500 descriptors configured:

```
                          sp_monitorconfig "aux scan descriptors"
Usage information at date and time: Jan 24 1997 9:54AM.
Name            # Free   # Active   % Active  # Max Ever Used   Re-used
--------------  --------  ------   --------   ---------------   -------
number of aux   260       240      48.00      427               NA
scan
descriptors
```

Only 240 auxiliary scan descriptors are being used, leaving 260 free. However, the maximum number of scan descriptors used at any one time since the last time Adaptive Server was started is 427, leaving about 20 percent for growth in use and exceptionally heavy use periods. “Re-used” does not apply to scan descriptors.

Estimating and configuring auxiliary scan descriptors

To get an estimate of scan descriptor use:

1. Determine the number of table references for any query referencing more than 16 user tables or those that have a large number of referential constraints, by running the query with set showplan and set noexec enabled. If auxiliary scan descriptors are required, showplan reports the number needed:

   ```
   Auxiliary scan descriptors required: 17
   ```

   The reported number includes all auxiliary scan descriptors required for the query, including those for all worker processes. If your queries involve only referential constraints, you can also use sp_helpconstraint, which displays a count of the number of referential constraints per table.

2. For each query that uses auxiliary scan descriptors, estimate the number of users who would run the query simultaneously and multiply. If 10 users are expected to run a query that requires 8 auxiliary descriptors, a total of 80 will be needed at any one time.

3. Add the per-query results to calculate the number of needed auxiliary scan descriptors.

number of mailboxes

Summary Information	
Name in pre-11.0 release	cnmbox
Default value	30

Summary Information	
Range of values	30–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

number of mailboxes specifies the number of mailbox structures allocated by Adaptive Server. Mailboxes, which are used in conjunction with messages, are used internally by Adaptive Server for communication and synchronization between kernel service processes. Mailboxes are not used by user processes. Do not modify this parameter unless instructed to do so by Sybase Technical Support.

number of messages

Summary Information	
Name in pre-11.0 release	cnmsg
Default value	64
Range of values	0–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

number of messages specifies the number of message structures allocated by Adaptive Server. Messages, which are used in conjunction with mailboxes, are used internally by Adaptive Server for communication and synchronization between kernel service processes. Messages are also used for coordination between a family of processes in parallel processing. Do not modify this parameter unless instructed to do so by Sybase Technical Support.

number of pre-allocated extents

Summary Information	
Name in pre-11.0 release	cpreallocext
Default value	2
Range of values	0–31
Status	Dynamic
Display level	Comprehensive

Summary Information	
Required role	System Administrator

number of pre-allocated extents specifies the number of extents (eight pages) allocated in a single trip to the page manager. Currently, it is used only by bcp to improve performance when copying in large amounts of data. By default, bcp allocates two extents at a time and writes an allocation record to the log each time.

Setting number of pre-allocated extents means that bcp allocates the specified number of extents each time it requires more space, and writes a single log record for the event. Setting the value to 0 disables extent allocation so that a single page is allocated each time bulk copy needs a page. Since each page allocation is logged, this can greatly increase the amount of transaction log space required.

An object may be allocated more pages than actually needed, so the value of number of pre-allocated extents should be low if you are using bcp for small batches. If you are using bcp for large batches, increase the value of number of pre-allocated extents to reduce the amount of overhead required to allocate pages and to reduce the number of log records.

number of sort buffers

Summary Information	
Name in pre-11.0 release	csortbufsize
Default value	500
Range of values	0–32767
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

number of sort buffers specifies the number of 2K buffers used to hold pages read from input tables and perform index merges during sorts.

Sybase recommends that you leave this parameter set to the default except when you are creating indexes in parallel. Setting the value too high can rob non-sorting processes of access to the 2K buffer pool in caches being used to perform sorts.

For more information on configuring this value for parallel create index statements, see “Caches, sort buffers, and parallel sorts” on page 591 in the *Performance and Tuning Guide*.

partition groups

Summary Information	
Name in pre-11.0 release	N/A
Default value	1024
Range of values	1–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

partition groups specifies the maximum number of partition groups that can be allocated by Adaptive Server. Partition groups are internal structures used by Adaptive Server to control access to individual partitions of a table.

A partition group is composed of 16 partition caches, each of which stores information about a single partition. All caches in a partition group are used to store information about the same partitioned table. If a table has fewer than 16 partitions, the unused partition caches in that group are unused, and cannot be used by another table. If a table has more than 16 partitions, it requires multiple partition groups.

The default value allows a maximum 1024 open partition groups and a maximum of 16384 (1024 times 16) open partitions. The actual number of partitions may be slightly less, due to the grouping of partitions.

Adaptive Server allocates partition groups to a table when you partition the table or when you access it for the first time after restarting Adaptive Server. If there are not enough partition groups for the table, you will not be able to access or partition the table.

partition spinlock ratio

Summary Information	
Name in pre-11.0 release	N/A
Default value	10
Range of values	1–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

For Adaptive Servers running with multiple engines, partition spinlock ratio sets the number of rows in the internal partition caches that are protected by one **spinlock**.

Adaptive Server manages access to table partitions using internal *partition groups*, each of which contains partition caches. Each partition cache stores information about a partition (for example, the last page of the partition) that processes must use when accessing that partition.

By default, Adaptive Server systems are configured with partition spinlock ratio set to 10, or 1 spinlock for every 10 partition caches. Decreasing the value of partition spinlock ratio may have little impact on the performance of Adaptive Server. The default setting is correct for most servers.

For more information about configuring spinlock ratios, see "Configuring spinlock ratio parameters" on page 644.

print deadlock information

Summary Information	
Name in pre-11.0 release	T1204 (trace flag)
Default value	0 (off)
Valid values	0 (off), 1 (on)
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

print deadlock information enables the printing of deadlock information to the error log.

If you are experiencing recurring deadlocks, setting print deadlock information to 1 provides you with information that can be useful in tracing the cause of the deadlocks. However, setting print deadlock information to 1 can seriously degrade Adaptive Server performance. For this reason, you should use it only when you are trying to determine the cause of deadlocks.

Use sp_sysmon output to determine whether deadlocks are occurring in your application. If they are, set print deadlock information to 1 to learn more about why they are occurring. See the *Performance and Tuning Guide* for more information.

runnable process search count

Summary Information	
Name in pre-11.0 release	cschedspins
Default value	2000
Range of values	0–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

runnable process search count specifies the number of times an engine loops while looking for a runnable task before relinquishing the CPU to the operating system.

Adaptive Server engines check the run queue for runnable tasks whenever a task completes or exceeds its allotted time on the engine. At times, there will not be any tasks in the run queues. An engine can either relinquish the CPU to the operating system or continue to check for a task to run. Setting runnable process search count higher causes the engine to loop more times, thus holding the CPU for a longer time. Setting the runnable process search count lower causes the engine to release the CPU sooner.

If your machine is a uniprocessor that depends on helper threads to perform I/O, you may see some performance benefit from setting runnable process search order to perform network I/O, disk I/O, or other operating system tasks. If a client, such as a bulk copy operation, is running on the same machine as a single CPU server that uses helper threads, it can be especially important to allow both the server and the client access to the CPU.

For Adaptive Servers running on uniprocessor machines that do not use helper threads, and for multiprocessor machines, the default value provides good performance.

Use sp_sysmon to determine how the runnable process search count parameter affects Adaptive Server's use of CPU cycles, engine yields to the operating system, and blocking network checks. See the *Performance and Tuning Guide* for information.

size of auto identity column

Summary Information	
Name in pre-11.0 release	N/A

Summary Information	
Default value	10
Range of values	1–38
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

size of auto identity column sets the precision of IDENTITY columns that are automatically created with the sp_dboption auto identity and unique auto_identity index options.

The maximum value that can be inserted into an IDENTITY column is $10^{precision}$ -1. After an IDENTITY column reaches its maximum value, all further insert statements return an error that aborts the current transaction.

If you reach the maximum value of an IDENTITY column, use the create table command to create a table that is identical to the old one, but with a larger precision for the IDENTITY column. After you have created the new table, use the insert command or bcp to copy data from the old table to the new one.

SQL Perfmon Integration (Windows NT only)

Summary Information	
Name in pre-11.0 release	N/A
Default value	1 (on)
Valid values	0 (off), 1 (on)
Status	Static
Display level	Intermediate
Required role	System Administrator

SQL Perfmon Integration enables and disables the ability to monitor Adaptive Server statistics from the Windows NT Performance Monitor.

Adaptive Server must be registered as an NT Service to support monitor integration. This occurs automatically when:

- You start Adaptive Server using the Services Manager in the Sybase for Windows NT program group.
- You use the Services option in the Control Panel.

- You have configured Windows NT to start Adaptive Server as an automatic service.

See *Configuring Adaptive Server for Windows NT* for a list of the Adaptive Server counters you can monitor.

sql server clock tick length

Summary Information	
Name in pre-11.0 release	cclkrate
Default value	Platform-specific
Range of values	Platform-specific minimum–1000000, in multiples of default value
Status	Static
Display level	Comprehensive
Required role	System Administrator

sql server clock tick length specifies the duration of the server's clock tick, in microseconds. Both the default value and the minimum value are platform-specific. Adaptive Server rounds values up to an even multiple of *n*, where *n* is the platform-specific clock-tick default value. You can find the current values for sql server clock tick length by using sp_helpconfig or sp_configure.

In mixed-use applications with some CPU-bound tasks, decreasing the value of sql server clock tick length helps I/O-bound tasks. A value of 20,000 is reasonable for this. Shortening the clock tick length means that CPU-bound tasks will exceed the allotted time on the engine more frequently per unit of time, which allows other tasks greater access to the CPU. This may also marginally increase response times, because Adaptive Server runs its service tasks once per clock tick. Decreasing the clock tick length means that the service tasks will be run more frequently per unit of time.

Increasing sql server clock tick length favors CPU-bound tasks, because they execute longer between context switches. The maximum value of 1,000,000 may be appropriate for primarily CPU-bound applications. However, any I/O-bound tasks may suffer as a result. This can be mitigated somewhat by tuning cpu grace time (see "cpu grace time" on page 181) and time slice (see "time slice" on page 208).

Note Changing the value of sql server clock tick length can have serious effects on Adaptive Server's performance. You should consult with Sybase Technical Support before resetting this value.

text prefetch size

Summary Information	
Name in pre-11.0 release	N/A
Default value	16
Valid values	0 to 65535
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

The text prefetch size parameter limits the number of pages of text and image data that can be prefetched into an existing buffer pool. Adaptive Server prefetches only text and image data that was created with Adaptive Server 12.x or was upgraded using dbcc rebuild_text.

time slice

Summary Information	
Name in pre-11.0 release	time slice
Default value	100
Range of values	50–1000
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

time slice sets the number of milliseconds that Adaptive Server's scheduler allows a task to run. If time slice is set too low, Adaptive Server may spend too much time switching between tasks, which increases response time. If it is set too high, CPU-intensive tasks might monopolize engines, which also increases response time. The default value, 100 milliseconds, allows each task to run for 1/10 of a second berfore relinquishing the CPU to another task.

See "cpu grace time" on page 181. Also, see Chapter 3, "Using Engines and CPUs," and "Execution task scheduling" on page 29 in the *Performance and Tuning Guide* for a more detailed discussion of task scheduling.

Use sp_sysmon to determine how time slice affects voluntary yields by Adaptive Server engines. See the *Performance and Tuning Guide* for more information.

upgrade version

Summary Information	
Name in pre-11.0 release	upgrade version
Default value	1100
Range of values	0–2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

upgrade version reports the version of the upgrade utility that upgraded your master device. The upgrade utility checks and modifies this parameter during an upgrade.

Warning! Although this parameter is configurable, you should not reset it. Doing so may cause serious problems with Adaptive Server.

You can determine whether an upgrade has been done on your master device by using upgrade version without specifying a value:

```
sp_configure "upgrade version"
```

row lock promotion HWM

Summary Information	
Default value	200
Range of values	2–2147483647
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

row lock promotion HWM (high-water mark), together with row lock promotion LWM (low-water mark) and row lock promotion PCT specifies the number of row locks permitted during a single scan session of a table or an index before Adaptive Server attempts to escalate from row locks to a table lock.

row lock promotion HWM sets a maximum number of row locks allowed on a table before Adaptive Server attempts to escalate to a table lock. When the number of locks acquired during a scan session exceeds row lock promotion HWM, Adaptive Server attempts to acquire a table lock. The lock promotion HWM value cannot be higher than the number of locks value.

For more information on scan sessions and setting up lock promotion limits, see "Configuring locks and lock promotion thresholds" on page 226in the *Performance and Tuning Guide*.

The default value for row lock promotion HWM is appropriate for most applications. You might want to raise the value to avoid table locking. For example, if you know that there are regular updates to 500 rows on a table that has thousands of rows, you can increase concurrency for the tables by setting row lock promotion HWM to around 500.

You can also configure row lock promotion at the per-object level. See sp_setrowlockpromote in the Adaptive Server Reference Manual.

row lock promotion LWM

Summary Information	
Default value	200
Range of values	2–value of row lock promotion HWM
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

row lock promotion LWM (low-water mark), together with the row lock promotion HWM (high-water mark) and row lock promotion PCT specifies the number of row locks permitted during a single scan session of a table or an index before Adaptive Server attempts to promote from row locks to a table lock.

row lock promotion LWM sets the number of locks below which Adaptive Server does not attempt to acquire a table lock on the object. The row lock promotion LWM must be less than or equal to row lock promotion HWM.

For more detailed information on scan sessions and setting up lock promotion limits, see "Configuring locks and lock promotion thresholds" on page 226 in the *Performance and Tuning Guide*.

The default value for row lock promotion LWM is sufficient for most applications. If Adaptive Server runs out of locks (except for an isolated incident), you should increase number of locks. See the *Performance and Tuning Guide* for more information.

You can also configure lock promotion at the per-object level. See sp_setrowlockpromote in the *Adaptive Server Reference Manual*.

row lock promotion PCT

Summary Information	
Default value	100
Range of values	1–100
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

If the number of locks held on an object is between row lock promotion LWM (low-water mark) and row lock promotion HWM (high-water mark), row lock promotion PCT sets the percentage of row locks (based on the number of rows in the table) above which Adaptive Server attempts to acquire a table lock.

For more information on setting up lock promotion limits, see "Configuring locks and lock promotion thresholds" on page 226 in the *Performance and Tuning Guide*.

The default value for row lock promotion PCT is appropriate for most applications.

You can also configure row lock promotion at the per-object level. See sp_setrowlockpromote in the *Adaptive Server Reference Manual*.

license information

Summary Information	
Default value	0
Valid values	$0–2^{31}$
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

license information allows Sybase System Administrators to monitor the number of user licenses used in Adaptive Server. Enabling this parameter only monitors the number of licenses issued; it does not enforce the license agreement.

If license information is set to 0, Adaptive Server does not monitor license use. If license information is set to a number greater than 0, the housekeeper task monitors the number of licenses used during the idle cycles in Adaptive Server. Set license information to the number of licenses specified in your license agreement.

If the number of licenses used is greater than the number to which license information is set, Adaptive Server writes the following error message to the error log:

```
WARNING: Exceeded configured number of user licenses
```

At the end of each 24-hour period, the maximum number of licenses used during that time is added to the syblicenseslog table. The 24-hour period restarts if Adaptive Server is restarted.

See "Monitoring license use" on page 382 for more information.

Security related

The parameters in this group configure security-related features.

allow procedure grouping

Summary Information	
Name in pre-11.0 release	N/A
Default value	1 (on)
Range of values	0 (off), 1 (on)
Status	Dynamic
Display level	Comprehensive
Required role	System Security Officer

allow procedure grouping controls the ability to group stored procedures of the same name so that they can be dropped with a single drop procedure statement. To run Adaptive Server in the *evaluated configuration*, you must prohibit stored procedure grouping by setting this option to 0. See **evaluated configuration** in the *Adaptive Server Glossary* for more information.

auditing

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (off)
Range of values	0 (off), 1 (on)
Status	Dynamic
Display level	Intermediate
Required role	System Security Officer

auditing enables or disables auditing for Adaptive Server.

audit queue size

Summary Information	
Name in pre-11.0 release	audit queue size
Default value	100
Range of values	1–65535
Status	Dynamic
Display level	Intermediate
Required role	System Security Officer

The in-memory audit queue holds audit records generated by user processes until the records can be processed and written to the audit trail. A System Security Officer can change the size of the audit queue with audit queue size. There is a trade-off between performance and risk that must be considered when you configure the queue size. If the queue is too large, records can remain in it for some time. As long as records are in the queue, they are at risk of being lost if the system crashes. However, if the queue is too small, it can become full repeatedly, which affects overall system performance–user processes that generate audit records sleep if the audit queue is full.

Following are some guidelines for determining how big your audit queue should be. You must also take into account the amount of auditing to be done at your site.

- The memory requirement for a single audit record is 424 bytes; however a record can be as small as 22 bytes when it is written to a data page
- The maximum number of audit records that can be lost in a system crash is the size of the audit queue (in records), plus 20. After records leave the audit queue they remain on a buffer page until they are written to the current audit table on the disk. The pages are flushed to disk every 20 records at the most (less if the audit process is not constantly busy).
- In the system audit tables, the extrainfo field and fields containing names are of variable length, so audit records that contain complete name information are generally larger.

The number of audit records that can fit on a page varies from 4 to as many as 80 or more. The memory requirement for the default audit queue size of 100 is approximately 42K.

current audit table

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Range of values	0–8
Status	Dynamic
Display level	Intermediate
Required role	System Security Officer

current audit table establishes the table where Adaptive Server writes audit rows. A System Security Officer can change the current audit table, using:

```
sp_configure "current audit table", n
  [, "with truncate"]
```

where n is an integer that determines the new current audit table, as follows:

- 1 means sysaudits_01, 2 means sysaudits_02, and so forth, up to 8.
- 0 tells Adaptive Server to set the current audit table to the next table. For example, if your installation has three audit tables, *sysaudits_01*, *sysaudits_02*, and *sysaudits_03*, Adaptive Server sets the current audit table to:
  - 2 if the current audit table is *sysaudits_01*
  - 3 if the current audit table is *sysaudits_02*
  - 1 if the current audit table is *sysaudits_03*

"with truncate" specifies that Adaptive Server should truncate the new table if it is not already empty. sp_configure fails if this option is not specified and the table is not empty.

Note If Adaptive Server truncates the current audit table, and you have not archived the data, the table's audit records are lost. Be sure that the audit data is archived before using the with truncate option.

To execute sp_configure to change the current audit table, you must have the sso_role active. You can write a threshold procedure to change the current audit table automatically.

enable ssl

Summary Information	
Name in pre-12.5 release	N/A
Default value	0
Valid values	0 (off), 1 (on)
Status	Static
Display level	Comprehensive
Required role	System Administrator

The enable ssl parameter enables or disables Secure Sockets Layer session-based security.

msg confidentiality reqd

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (off)
Range of values	0 (off), 1 (on)
Status	Dynamic
Display level	Intermediate
Required role	System Security Officer

The msg confidentiality reqd parameter requires that all messages into and out of Adaptive Server be encrypted. The use security services parameter must be 1 for messages to be encrypted.

msg integrity reqd

Summary Information	
Name in pre-11.0 release	N/A
Default value	0 (off)
Range of values	0 (off), 1 (on)
Status	Dynamic
Display level	Intermediate
Required role	System Security Officer

msg integrity reqd requires that all messages be checked for data integrity. use security services must be 1 for message integrity checks to occur. If msg integrity reqd is set to one, Adaptive Server allows the client connection to succeed unless the client is using one of the following security services: message integrity, replay detection, origin checks, or out-of-seq checks.

secure default login

Summary Information	
Name in pre-11.0 release	N/A

Summary Information	
Default value	0
Range of values	0 (followed by another parameter naming the default login)
Status	Dynamic
Display level	Intermediate
Required role	System Security Officer

secure default login specifies a default login for all users who are preauthenticated but who do not have a login in master..syslogins.

Establish the secure default login with:

```
sp_configure "secure default login", 0,
    default_login_name
```

where:

- secure default login is the name of the parameter.
- 0 is a required parameter because the second parameter of sp_configure must be a numeric value.
- *default_login_name* is the name of the default login for a user who is unknown to Adaptive Server, but who has already been authenticated by a security mechanism. The login name must be a valid login in master..syslogins.

For example, to specify "dlogin" as the secure default login, type:

```
sp_configure "secure default login", 0, dlogin
```

select on syscomments.text column

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Range of values	0–1
Status	Dynamic
Display level	Comprehensive
Required role	System Security Officer

This parameter enables protection of the text of database objects through restriction of the select permission on the text column of the syscomments table. The default value of 1 allows select permission to "public." Set the option to 0 to restrict select permission to the object owner and the System Administrator.

To run Adaptive Server in the *evaluated configuration*, you must protect the source text of database objects by setting this option to 0.

See **evaluated configuration** in the *Adaptive Server Glossary* for more information.

suspend audit when device full

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Range of values	0–1
Status	Dynamic
Display level	Intermediate
Required role	System Security Officer

suspend audit when device full determines what Adaptive Server does when an audit device becomes completely full.

Note If you have two or more audit tables, each on a separate device other than the master device, and you have a threshold procedure for each audit table segment, the audit devices should never become full. Only if a threshold procedure is not functioning properly would the "full" condition occur.

Choose one of these values:

- 0 – truncates the next audit table and starts using it as the current audit table when the current audit table becomes full. If you set the parameter to 0, you ensure that the audit process is never suspended. However, you incur the risk that older audit records will get lost if they have not been archived.

- 1 – suspends the audit process and all user processes that cause an auditable event. To resume normal operation, the System Security Officer must log in and set up an empty table as the current audit table. During this period, the System Security Officer is exempt from normal auditing. If the System Security Officer's actions would generate audit records under normal operation, Adaptive Server sends an error message and information about the event to the error log.

To run in the evaluated configuration, set this parameter to 1. See **evaluated configuration** in the *Adaptive Server Glossary* for more information.

systemwide password expiration

Summary Information	
Name in pre-11.0 release	password expiration interval
Default value	0
Range of values	0–32767
Status	Dynamic
Display level	Intermediate
Required role	System Security Officer

systemwide password expiration, which can be set only by a System Security Officer, sets the number of days that passwords remain in effect after they are changed. If systemwide password expiration is set to 0, passwords do not expire. If it is set to a number greater than 0, all passwords expire after the specified number of days. An account's password is considered expired if an interval greater than *number_of_days* has passed since the last time the password for that account was changed.

When the number of days remaining before expiration is less than 25 percent of the value of systemwide password expiration or 7 days, whichever is greater, each time the user logs in, a message displays, giving the number of days remaining before expiration. Users can change their passwords anytime before expiration.

When an account's password has expired, the user can still log in to Adaptive Server but cannot execute any commands until he or she has used sp_password to change his or her password. If the System Security Officer changes the user's password while the account is in sp_password-only mode, the account returns to normal after the new password is assigned.

This restriction applies only to login sessions established after the password has expired. Users who are logged in at the time their passwords expire are not affected until the next time they log in.

unified login required (Windows NT only)

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0, 1
Status	Dynamic
Display level	Intermediate
Required role	System Security Officer

unified login required requires that all users who log in to Adaptive Server be authenticated by the Windows NT LAN Manager. The use security services parameter must be 1 to use the unified login security service.

use security services (Windows NT only)

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0, 1
Status	Static
Display level	Intermediate
Required role	System Security Officer

use security services specifies that Adaptive Server will use security services provided by Windows NT LAN Manager. If the parameter is set to 0, unified login services with the LAN Manager cannot be used.

Unicode

The parameters in this group configure Unicode-related features.

default unicode sortorder

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	(not currently used)
Status	Static
Display level	Comprehensive
Required role	System Administrator

The default unicode sortorder parameter is a string parameter that defines the default Unicode sort order installed on the server. A string parameter is used rather than a numeric parameter to guarantee a unique id. To change the Unicode default sort order, see Chapter 7, "Configuring Character Sets, Sort Orders, and Languages."

enable surrogate processing

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Range of values	0 – 1
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

Activates the processing and maintains the integrity of surrogate pairs in Unicode data. Set enable surrogate processing to 1 to enable surrogate processing. If this is disabled, the server ignores the presence of surrogate pairs in the Unicode data, and all code that maintains the integrity of surrogate pairs is skipped. This enhances performance, but restricts the range of Unicode characters that can appear in the data.

enable unicode conversion

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0 – 2

Summary Information	
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

Activates character conversion using Unilib for the char, varchar, and text datatypes. Set enable unicode conversion to 1 to use the built-in conversion. If it can't find a built-in conversion, Adaptive Server uses the Unilib character conversion. Set enable unicode conversion to 2 to use the appropriate Unilib conversion. Set the parameter to 0 to use only the built-in character-set conversion.

enable unicode normalization

Summary Information	
Name in pre-11.0 release	N/A
Default value	1
Range of values	0 – 1
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

Activates Unilib character normalization. The normalization process modifies the data so there is only a single representation in the database for a given sequence of abstract characters. Often, characters folllowed by combined diacritics are replaced by pre-combined forms.

Set enable unicode normalization to 1 to use the built-in process that enforces normalization on all incoming Unicode data. If this parameter is disabled (set to 0), the normalization step is bypassed and the client code is responsible for normalization rather than the server. If normalization is disabled, performance is improved—but only if *all* clients present Unicode data to the server using the same representation.

Note Once disabled, normalization cannot be turned on again. This one-way change, prevents non-normalized data from entering the data base.

size of unilib cache

Summary Information	
Name in pre-11.0 release	N/A
Default value	0
Range of values	0 – 2147483647
Status	Dynamic
Display level	Comprehensive
Required role	System Administrator

Determines the size of the Unilib cache. size of unilib cache specifies the size in bytes. You may need a larger cache if your site uses multiple conversions.

User environment

The parameters in this group configure user environments.

number of user connections

Summary Information	
Name in pre-11.0 release	user connections
Default value	25
Range of values	5–2147483647
Status	Dynamic
Display level	Basic
Required role	System Administrator

number of user connections sets the maximum number of user connections that can be connected to Adaptive Server at the same time. It does not refer to the maximum number of processes; that number depends not only on the value of this parameter but also on other system activity.

Upper Limit to the *maximum number of user connections*

The maximum allowable number of file descriptors per process is operating-system-dependent; see the configuration documentation for your platform.

The number of file descriptors available for Adaptive Server connections is stored in the global variable *@@max_connections*. You can report the maximum number of file descriptors your system can use with:

```
select @@max_connections
```

The return value represents the maximum number of file descriptors allowed by the system for your processes, minus overhead. Overhead increases with the number of engines. For more information on how multiprocessing affects the number file descriptors available for Adaptive Server connections, see "Managing user connections" on page 643.

In addition, you must reserve a number of connections for the following items, which you also set with configuration parameters:

- The database devices, including mirror devices
- Site handlers
- Network listeners

The following formula determines how high you can set number of user connections, number of devices, max online engines, number of remote sites, and max number network listeners:

number of user connections + (number of devices * max online engines * 2) + number of remote sites + max number network listeners cannot be greater than the value of *@@max_connections*.

Optimizing the Value of the *max number of user connections* Parameter

There is no formula for determining how many connections to allow for each user. You must estimate this number, based on the system and user requirements described here. You must also take into account that on a system with many users, there is more likelihood that connections needed only occasionally or transiently can be shared among users. The following processes require user connections:

- One connection is needed for each user running isql.
- Application developers use one connection for each editing session.

- The number of connections required by users running an application depends on how the application has been programmed. Users executing Open Client programs need one connection for each open DB-Library dbprocess or Client-Library cs_connection.

Note It is a good idea to estimate the maximum number of connections that will be used by Adaptive Server and to update number of user connections as you add physical devices or users to the system. Use sp_who periodically to determine the number of active user connections on your Adaptive Server.

Certain other configuration parameters, including stack size and default network packet size, affect the amount of memory for each user connection.

User connections for shared memory

Adaptive Server uses the value of the number of user connections parameter to establish the number of shared-memory connections for EJB Server. Thus, if number of user connections is 30, Adaptive Server establishes 10 shared-memory connections for EJB Server. Shared-memory connections are not a subset of user connections, and are not subtracted from the number of user connections.

To increase the number of user connections for shared memory, you must:

1 Increase number of user connections to a number one-third of which is the number of desired shared-memory connections.

2 Reboot Adaptive Server.

Although number of user connections is a dynamic configuration parameter, you must restart the server to change the number of user connections for shared memory. See the *EJB Server User's Guide* for more information.

permission cache entries

Summary Information	
Name in pre-11.0 release	cfgcprot
Default value	15
Range of values	1–2147483647
Status	Dynamic
Display level	Comprehensive

Summary Information	
Required role	System Administrator

permission cache entries determines the number of cache protectors per task. This parameter increases the amount of memory for each user connection and worker process.

Information about user permissions is held in the permission cache. When Adaptive Server checks permissions, it looks first in the permission cache; if it does not find what it needs, it looks in the sysprotects table. It is significantly faster if Adaptive Server finds the information it needs in the permission cache and does not have to read sysprotects.

However, Adaptive Server looks in the permission cache only when it is checking user permissions, not when permissions are being granted or revoked. When a permission is granted or revoked, the entire permission cache is flushed. This is because existing permissions have timestamps that become outdated when new permissions are granted or revoked.

If users on your Adaptive Server frequently perform operations that require their permissions to be checked, you may see a small performance gain by increasing the value of permission cache entries. This effect is not likely to be significant enough to warrant extensive tuning.

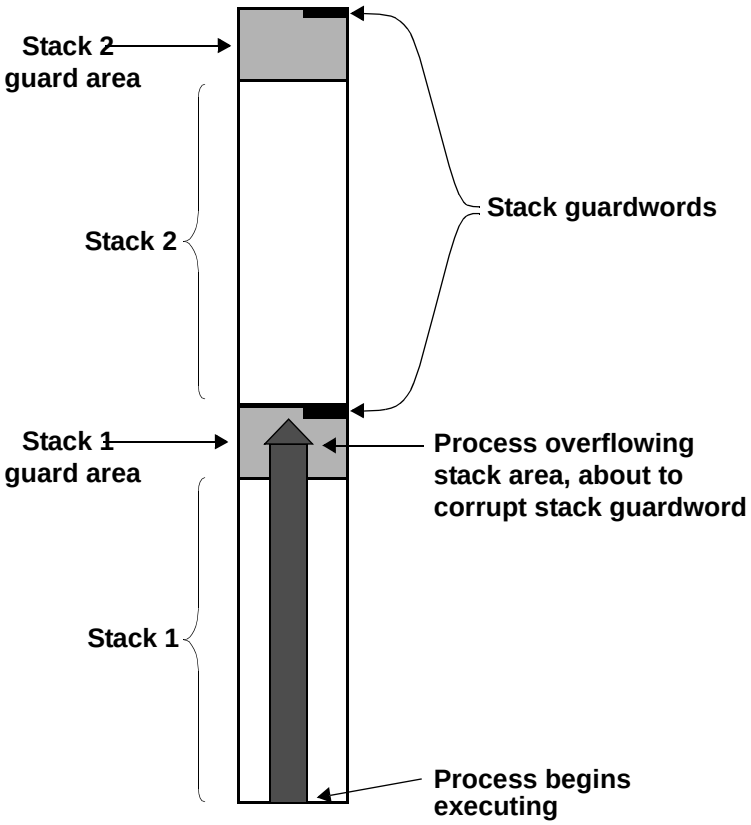
If users on your Adaptive Server frequently grant or revoke permissions, avoid setting permission cache entries to a large value. The space used for the permission cache would be wasted, since the cache is flushed with each grant and revoke command.

stack guard size

Summary Information	
Name in pre-11.0 release	cguardsz
Default value	4096
Range of values	0–2147483647
Status	Static
Display level	Comprehensive
Required role	System Administrator

stack guard size sets the size (in bytes) of the stack guard area. The *stack guard area* is an overflow stack of configurable size at the end of each stack. Adaptive Server allocates one stack for each user connection and worker process when it starts. These stacks are located contiguously in the same area of memory, with a guard area at the end of each stack. At the end of each stack guard area is a *guardword*, which is a 4-byte structure with a known pattern. Figure 5-7 illustrates how a process can corrupt a stack guardword.

Figure 5-7: Process about to corrupt stack guardword



Adaptive Server periodically checks to see whether the stack pointer for a user connection has entered the stack guard area associated with that user connection's stack. If it has, Adaptive Server aborts the transaction, returns control to the application that generated the transaction, and generates Error 3626:

```
The transaction was aborted because it used too much
stack space.  Either use sp_configure to increase the
stack size, or break the query into smaller pieces.
spid: %d, suid: %d, hostname: %.*s, application
name: %.*s
```

Adaptive Server also periodically checks the guardword pattern to see if it has changed, thus indicating that a process has overflowed the stack boundary. When this occurs, Adaptive Server prints these messages to the error log and shuts down:

```
kernel: *** Stack overflow detected: limit: 0x%lx sp: 0x%lx
kernel: *** Stack Guardword corrupted
kernel: *** Stack corrupted, server aborting
```

In the first message, "limit" is the address of the end of the stack guard area, and "sp" is the current value of the stack pointer.

In addition, Adaptive Server periodically checks the stack pointer to see whether it is completely outside both the stack and the stack guard area for the pointer's process. If it is, Adaptive Server shuts down, even if the guardword is not corrupted. When this happens, Adaptive Server prints the following messages to the error log:

```
kernel: *** Stack overflow detected: limit: 0x%lx sp: 0x%lx
kernel: *** Stack corrupted, server aborting
```

The default value for stack guard size is appropriate for most applications. However, if you experience server shutdown from either stack guardword corruption or stack overflow, increase stack guard size by a 2K increment. *Each* configured user connection and worker process has a stack guard area; thus, when you increase stack guard size, you use up that amount of memory, multiplied by the number of user connections and worker processes you have configured.

Rather than increasing stack guard size to avoid stack overflow problems, consider increasing stack size (see "stack size" on page 229). The stack guard area is intended as an overflow area, not as an extension to the regular stack.

Adaptive Server allocates stack space for each task by adding the values of the stack size and stack guard size parameters. stack guard size must be configured in multiples of 2K. If the value you specify is not a multiple of 2K, sp_configure verification routines round the value up to the next highest multiple.

stack size

Summary Information	
Name in pre-11.0 release	stack size
Default value	platform-specific
Range of values	Platform-specific minimum–2147483647
Status	Static
Display level	Basic
Required role	System Administrator

stack size specifies the size (in bytes) of the execution stacks used by each user process on Adaptive Server. To find the stack size values for your platform, use sp_helpconfig or sp_configure. stack size must be configured in multiples of 2K. If the value you specify is not a multiple of 2K, sp_configure verification routines round the value up to the next highest multiple.

An *execution stack* is an area of Adaptive Server memory where user processes keep track of their process context and store local data.

Certain queries can contribute to the probability of a stack overflow. Examples include queries with extremely long where clauses, long select lists, deeply nested stored procedures, and multiple selects and updates using holdlock. When a stack overflow occurs, Adaptive Server prints an error message and rolls back the transaction. See "stack guard size" on page 226 for more information on stack overflows. See the *Troubleshooting and Error Messages Guide* for more information on specific error messages.

The two options for remedying stack overflows are to break the large queries into smaller queries and to increase stack size. Changing stack size affects the amount of memory required for *each* configured user connection and worker process. See "total logical memory" on page 173 for further information.

If you have queries that exceed the size of the execution stack, you may want to rewrite them as a series of smaller queries. This is particularly true if there are only a small number of such queries or if you run them infrequently.

There is no way to determine how much stack space a query will require without actually running the query. Stack space for each user connection and worker process is preallocated at start-up.

Therefore, determining the appropriate value for stack size is an empirical process. You should test your largest and most complex queries using the default value for stack size. If they run without generating error messages, the default is probably sufficient. If they generate error messages, you should begin by increasing stack size by a small amount (2K). Rerun your queries and see if the amount you have added is sufficient. If it is not, continue to increase stack size until queries run without generating error messages.

If you are using CIS, or if Java is enabled in the database and you want to use methods that call JDBC, Sybase recommends that you increase the default by 50 percent. If you are not using JDBC or CIS, the standard default value is sufficient.

user log cache size

Summary Information	
Name in pre-11.0 release	N/A
Default value	Logical page size
Range of values	2048[a] – 2147483647 a. Minimum determined by server's logical page size
Status	Static
Display level	Intermediate
Required role	System Administrator

user log cache size specifies the size (in bytes) for each user's log cache. It's size is determined by the server's logical page size. There is one user log cache for each configured user connection and worker process. Adaptive Server uses these caches to buffer the user transaction log records, which reduces the contention at the end of the transaction log.

When a user log cache becomes full or another event occurs (such as when the transaction completes), Adaptive Server "flushes" all log records from the user log cache to the database transaction log. By first consolidating the log records in each user's log cache, rather than immediately adding each record to the database's transaction log, Adaptive Server reduces contention of processes writing to the log, especially for SMP systems configured with more than one engine.

Note For transactions using a database with mixed data and log segments, the user log cache is flushed to the transaction log after each log record. No buffering takes place. If your databases do not have dedicated log segments, you should not increase the user log cache size.

Do not configure user log cache size to be larger than the maximum amount of log information written by an application's transaction. Since Adaptive Server flushes the user log cache when the transaction completes, any additional memory allocated to the user log cache is wasted. If no transaction in your server generates more than 4000 bytes of transaction log records, set user log cache size no higher than that value. For example:

```
sp_configure "user log cache size", 4000
```

Setting user log cache size too high wastes memory. Setting it too low can cause the user log cache to fill up and flush more than once per transaction, increasing the contention for the transaction log. If the volume of transactions is low, the amount of contention for the transaction log may not be significant.

Use sp_sysmon to understand how this parameter affects cache behavior. See the *Performance and Tuning Guide* for more information.

user log cache spinlock ratio

Summary Information	
Name in pre-11.0 release	N/A
Default value	20
Range of values	1–2147483647
Status	Dynamic
Display level	Intermediate
Required role	System Administrator

For Adaptive Servers running with multiple engines, the user log cache spinlock ratio parameter specifies the ratio of user log caches per user log cache **spinlock**. There is one user log cache for each configured user connection.

The default value for this parameter is 20, or one spinlock for each 20 user connections configured for your server.

Use sp_sysmon to understand how this parameter affects cache behavior. See the *Performance and Tuning Guide* for more information.

For more information about configuring spinlock ratios, see "Configuring spinlock ratio parameters" on page 644.

CHAPTER 6

Limiting Access to Server Resources

This chapter describes how to use resource limits to restrict the I/O cost, row count, or processing time that an individual login or application can use during critical times. It also describes how to create named time ranges to specify contiguous blocks of time for resource limits.

Topics covered in this chapter include:

What are resource limits?

Adaptive Server provides resource limits to help System Administrators prevent queries and transactions from monopolizing server resources. A *resource limit* is a set of parameters specified by a System Administrator to prevent an individual login or application from:

- Exceeding estimated or actual I/O costs, as determined by the optimizer
- Returning more than a set number of rows
- Exceeding a given elapsed time

The set of parameters for a resource limit includes the time of day to enforce the limit and the type of action to take. For example, you can prevent huge reports from running during critical times of the day, or kill a session whose query produces unwanted **Cartesian products**.

Planning resource limits

In planning a resource limit, consider:

- When to impose the limit (times of day and days of the week)
- Which users and applications to monitor
- What type of limit to impose
    - I/O cost (estimated or actual) for queries that may require large numbers of logical and physical reads
    - Row count for queries that may return large result sets
    - Elapsed time for queries that may take a long time to complete either because of their own complexity or because of external factors such as server load
- Whether to apply a limit to individual queries or to specify a broader scope (query batch or transaction)
- Whether to enforce the I/O cost limits prior to or during execution
- What action to take when the limit is exceeded (issue a warning, abort the query batch or transaction, or kill the session)

After completing the planning, use system procedures to:

- Specify times for imposing the limit by creating a named time range using sp_add_time_range
- Create new resource limits using sp_add_resource_limit
- Obtain information about existing resource limits using sp_help_resource_limit
- Modify time ranges and resource limits using sp_modify_time_range and sp_modify_resource_limit, respectively
- Drop time ranges and resource limits using sp_drop_time_range and sp_drop_resource_limit, respectively

Enabling resource limits

Configure Adaptive Server to enable resource limits. Use allow resource limits configuration parameter:

```
sp_configure "allow resource limits", 1
```

1 enables the resource limits; 0 disables them. allow resource limits is static, so you must restart the server to reset the changes.

allow resource limits signals the server to allocate internal memory for time ranges, resource limit, s and internal server alarms. It also internally assigns applicable ranges and limits to login sessions.

Setting allow resource limits to 1 also changes the output of showplan and statistics i/o, as follows:

- showplan displays estimated I/O cost information for DML statements. The information displayed is the optimizer's cost estimate for the query as a unitless number. The total estimated I/O cost is displayed for the query as a whole. This cost estimate is dependent on the table statistics (number and distribution of values) and the size of the appropriate buffer pools. It is independent of such factors as the state of the buffer pools and the number of active users. For more information, see "Messages describing access methods, caching, and I/O cost" on page 793 in the *Performance and Tuning Guide*.
- statistics io includes the actual total I/O cost of a statement according to the optimizer's costing formula. This value is a number representing the sum of the number of logical I/Os multiplied by the cost of a logical I/O and the number of physical I/Os multiplied by the cost of a physical I/O. For more information on these numbers, see "How Is "Fast" Determined?" in the *Performance and Tuning Guide*.

Defining time ranges

A *time range* is a contiguous block of time within a single day across one or more contiguous days of the week. It is defined by its starting and ending periods.

Adaptive Server includes predefined “at all times” range, which covers the period midnight through midnight, Monday through Sunday. You can create, modify, and drop additional time ranges as necessary for resource limits.

Named time ranges may overlap. However, the limits for a particular user/application combination may not be associated with named time ranges that overlap. You can create different limits that share the same time range.

For example, assume that you limit “joe_user” to returning 100 rows when he is running the payroll application during business hours. Later, you attempt to limit his row retrieval during peak hours, which overlap with business hours. You will get a message that the new limit failed, because it would have overlapped with an existing limit.

Although you cannot limit the row retrieval for “joe_user” in the payroll application during overlapping time ranges, nothing stops you from putting a second limit on “joe_user” during the same time range as the row retrieval limit. For example, you can limit the amount of time one of his queries can run to the same time range that you used to limit his row retrieval.

When you create a named time range, Adaptive Server stores it in the systimeranges system table to control when a resource limit is active. Each time range has a range ID number. The “at all times” range is range ID 1. Adaptive Server messages refer to specific time ranges.

Determining the time ranges you need

Use a chart like the one below to determine the time ranges to create for each server. Monitor server usage throughout the week; then indicate the periods when your server is especially busy or is performing crucial tasks that should not be interrupted.

Day / Time	00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	00:00
Mon																									
Tues																									
Wed																									
Thurs																									
Fri																									
Sat																									
Sun																									

Creating named time ranges

Create new time ranges use sp_add_time_range to:

- Name the time range
- Specify the days of the week to begin and end the time range
- Specify the times of the day to begin and end the time range

For syntax and detailed information, see sp_add_time_range in the *Adaptive Server Reference Manual*.

A time range example

Assume that two critical jobs are scheduled to run every week at the following times.

- Job 1 runs from 07:00 to 10:00 on Tuesday and Wednesday.

- Job 2 runs from 08:00 on Saturday to 13:00 on Sunday.

The following table uses “1” to indicate when job 1 runs and “2” to indicate when job 2 runs:

Day / Time	00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	00:00
Mon																									
Tues								1	1	1	1														
Wed								1	1	1	1														
Thurs																									
Fri																									
Sat									2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Sun	2	2	2	2	2	2	2	2	2	2	2	2	2	2											

Job 1 can be covered by a single time range, tu_wed_7_10:

```
sp_add_time_range tu_wed_7_10, tuesday, wednesday, "7:00", "10:00"
```

Job 2, however, requires two separate time ranges, for Saturday and Sunday:

```
sp_add_time_range saturday_night, saturday, saturday, "08:00", "23:59"
sp_add_time_range sunday_morning, sunday, sunday, "00:00", "13:00"
```

Modifying a named time range

Use sp_modify_time_range to:

- Specify which time range to modify
- Specify the change to the days of the week
- Specify the change to the times of the day

For syntax and detailed information, see sp_modify_time_range in the *Adaptive Server Reference Manual*.

For example, to change the end day of the *business_hours* time range to Saturday, retaining the existing start day, start time, and end time, enter:

```
sp_modify_time_range business_hours, NULL, Saturday, NULL, NULL
```

To specify a new end day and end time for the *before_hours* time range, enter:

```
sp_modify_time_range before_hours, NULL, Saturday, NULL, "08:00"
```

Note You cannot modify the "at all times" time range.

Dropping a named time range

Use sp_drop_time_range to drop a user-defined time range

For syntax and detailed information, see sp_drop_time_range in the *Adaptive Server Reference Manual*.

For example, to remove the *evenings* time range from the systimeranges system table in the master database, enter:

```
sp_drop_time_range evenings
```

Note You cannot drop the "at all times" time range or any time range for which resource limits are defined.

When do time range changes take effect?

The active time ranges are bound to a login session at the beginning of each query batch. A change in the server's active time ranges due to a change in actual time has no effect on a session during the processing of a query batch. In other words, if a resource limit restricts query batches during a given time range, but the query batch begins before that time range becomes active, the query batch that is already running is not affected by the resource limit. However, if you run a second query batch during the same login session, that query batch will be affected by the change in time.

Adding, modifying, and deleting time ranges does not affect the active time ranges for the login sessions currently in progress.

If a resource limit has a transaction as its scope, and a change occurs in the server's active time ranges while a transaction is running, the newly active time range does not affect the transaction currently in progress.

Identifying users and limits

For each resource limit, you must specify the object to which the limit applies.

You can apply a resource limit to any of the following:

- All applications used by a particular login
- All logins that use a particular application
- A specific application used by a particular login

where *application* is defined as a client program running on top of Adaptive Server, accessed through a particular login. To run an application on Adaptive Server, you must specify its name through the CS_APPNAME connection property using cs_config (an Open Client Client-Library application) or the DBSETLAPP function in Open Client DB-Library. To list named applications running on your server, select the program_name column from the master..sysprocesses table.

For more information about the CS_APPNAME connection property, see the *Open Client Client-Library/C Reference Manual*. For more information on the DBSETLAPP function, see the *Open Client DB-Library/C Reference Manual*.

Identifying heavy-usage users

Before you implement resource limits, run sp_reportstats. The output from this procedure will help you identify the users with heavy system usage. For example:

```
                sp_reportstats
Name    Since         CPU     Percent CPU  I/O     Percent I/O
------  -----------   -----   ------------ -----   -------------
probe   jun 19 1993   0       0%           0       0%
julie   jun 19 1993   10000   24.9962%     5000    24.325%
jason   jun 19 1993   10002   25.0013%     5321    25.8866%
```

```
ken     jun 19 1993   10001   24.9987%     5123    24.9234%
kathy   jun 19 1993   10003   25.0038%     5111    24.865%
                           Total CPU   Total I/O
                           ---------   ---------
                           40006       20555
```

The output above indicates that usage is balanced among the users. For more information on chargeback accounting, see "cpu accounting flush interval" on page 180 and "i/o accounting flush interval" on page 192.

Identifying heavy-usage applications

To identify the applications running on your system and the users who are running them, query the sysprocesses system table in the master database.

The following query determines that isql, payroll, perl, and acctng are the only client programs whose names were passed to the Adaptive Server:

```
select spid, cpu, physical_io,
  substring(user_name(uid),1,10) user_name,
  hostname, program_name, cmd
from sysprocesses
```

```
spid  cpu    physical_io  user_name hostname program_name cmd
----  ---    -----------  --------- -------- ------------ ------
  17    4          12748  dbo       sabrina  isql         SELECT
 424    5              0  dbo       HOWELL   isql         UPDATE
 526    0            365  joe       scotty   payroll      UPDATE
 568    1           8160  dbo       smokey   perl         SELECT
 595   10              1  dbo       froth    isql         DELETE
 646    1              0  guest     walker   isql         SELECT
 775    4          48723  joe_user  mohindra acctng       SELECT

(7 rows affected)
```

Because sysprocesses is built dynamically to report current processes, repeated queries produce different results. Repeat this query throughout the day over a period of time to determine which applications are running on your system.

The CPU and physical I/O values are flushed to the syslogins system table periodically where they increment the values shown by sp_reportstats.

After identifying the applications running on your system, use showplan and statistics io to evaluate the resource usage of the queries in the applications.

If you have configured Adaptive Server to enable resource limits, you can use showplan to evaluate resources used prior to execution and statistics io to evaluate resources used during execution. For information on configuring Adaptive Server to enable resource limits, see "Enabling resource limits" on page 235.

In addition to statistics io, statistics time is also useful for evaluating the resources a query consumes. Use statistics time to display the time it takes to execute each step of the query. For more information, see "Diagnostic Tools for Query Optimization" on page 12-6 in the *Performance and Tuning Guide*.

Choosing a limit type

After you determine the users and applications to limit, you have a choice of three different types of resource limits.

Table 6-1 describes the function and scope of each limit type and indicates the tools that help determine whether a particular query might benefit from this type of limit. In some cases, it may be appropriate to create more than one type of limit for a given user and application. For more information on limit types, see "Understanding limit types" on page 245.

Table 6-1: Resource limit types

Limit type	Use for queries that	Measuring resource usage	Scope	Enforced during
io_cost	Require many logical and physical reads	Use set showplan on before running the query, to display its estimated I/O cost; use set statistics io on to observe the actual I/O cost.	Query	Pre-execution or execution
row_count	Return large result sets	Use the *@@rowcount* global variable to help develop appropriate limits for row count.	Query	Execution
elapsed_time	Take a long time to complete, either because of their own complexity or because of external factors such as server load or waiting for a lock	Use set statistics time on before running the query, to display elapsed time in milliseconds.	Query batch or transaction	Execution

The spt_limit_types system table stores information about each limit type.

Determining time of enforcement

Time of enforcement is the phase of query processing during which Adaptive Server applies a given resource limit. Resource limits occur during:

- Pre-execution – Adaptive Server applies resource limits prior to execution, based on the optimizer's I/O cost estimate. This limit prevents execution of potentially expensive queries. I/O cost is the only resource type that can be limited at pre-execution time.

  When evaluating the I/O cost of data manipulation language (DML) statements within the clauses of a conditional statement, Adaptive Server considers each DML statement individually. It evaluates all statements, even though only one clause will actually be executed.

  A pre-execution time resource limit can have only a query limit scope; that is, the values of the resources being limited at compile time are computed and monitored on a query-by-query basis only.

  Adaptive Server does not enforce pre-execution time resource limits statements in a trigger.

- Execution – Adaptive Server applies resource limits at runtime, and is usually used to prevent a query from monopolizing server and operating system resources. Execution time limits may use more resources (additional CPU time as well as I/O) than pre-execution time limits.

Determining the scope of resource limits

The *scope* parameter specifies the duration of a limit in Transact-SQL statements. The possible limit scopes are query, query batch, and transaction:

- Query – Adaptive Server applies resource limits to any single Transact-SQL statement that accesses the server; for example, select, insert, and update. When you issue these statements within a query batch, Adaptive Server evaluates them individually.

  Adaptive Server considers a stored procedure to be a series of DML statements. It evaluates the resource limit of each statement within the stored procedure. If a stored procedure executes another stored procedure, Adaptive Server evaluates each DML statement within the nested stored procedure at the inner nesting level.

Adaptive Server checks pre-execution time resource limits with a query scope, one nesting level at a time. As Adaptive Server enters each nesting level, it checks the active resource limits against the estimated resource usage of each DML statement prior to executing any of the statements at that nesting level. A resource limit violation occurs if the estimated resource usage of any DML query at that nesting level exceeds the limit value of an active resource limit. Adaptive Server takes the action that is bound to the violated resource limit.

Adaptive Server checks execution time resource limits with a query scope against the cumulative resource usage of each DML query. A limit violation occurs when the resource usage of a query exceeds the limit value of an active execution time resource limit. Again, Adaptive Server takes the action that is bound to that resource limit.

- Query batch – query batch consists of one or more Transact-SQL statements; for example, in isql, a group of queries becomes a query batch when executed by a single go command terminator.

  The query batch begins at nesting level 0; each call to a stored procedure increments the nesting level by 1 (up to the maximum nesting level). Each return from a stored procedure decrements the nesting level by 1.

  Only execution time resource limits can have a query batch scope.

  Adaptive Server checks execution time resource limits with a query batch scope against the cumulative resource usage of the statements in each query batch. A limit violation occurs when the resource usage of the query batch exceeds the limit value of an active execution time resource limit. Adaptive Server takes the action that is bound to that resource limit.

- Transaction – Adaptive Server applies limits with a transaction scope to all nesting levels during the transaction against the cumulative resource usage for the transaction.

  A limit violation occurs when the resource usage of the transaction exceeds the limit value of an active execution time resource limit. Adaptive Server takes the action that is bound to that resource limit.

  Only execution time resource limits can have a transaction scope.

Adaptive Server does not recognize nested transactions when applying resource limits. A resource limit on a transaction begins when *@@trancount* is set to 1 and ends when *@@trancount* is set to 0.

Understanding limit types

There are three types of resource limits that allow you to limit resource usage in different ways.

Limiting I/O cost

I/O cost is based on the number of logical and physical accesses ("reads") used during query processing. To determine the most efficient processing plan prior to execution, the Adaptive Server optimizer uses both logical and physical resources to compute an estimated I/O cost.

Adaptive Server uses the result of the optimizer's costing formula as a "unitless" number; that is, a value not necessarily based on a single unit of measurement (such as seconds or milliseconds).

To set resource limits, you must understand how those limits translate into runtime system overhead. For example, you must know the effect that a query with a cost of *x* logical and of *y* physical I/Os has on a production server.

Limiting io_cost can control I/O intensive queries, including queries that return a large result set. However, if you run a simple query that returns all the rows of a large table, and you do not have current statistics on the table's size, the optimizer may not estimate that the query will exceed the io_cost resource limit. To prevent queries from returning large result sets, create a resource limit on row_count.

The tracking of I/O cost limits may be less precise for partitioned tables than for unpartitioned tables when Adaptive Server is configured for parallel query processing. For more information on using resource limits in parallel queries, see the *Performance and Tuning Guide*.

Identifying I/O costs

To develop appropriate limits for I/O cost, determine the number of logical and physical reads required for some typical queries. Use the following set commands:

- set showplan on displays the optimizer's cost estimate. Use this information to set pre-execution time resource limits. A pre-execution time resource limit violation occurs when the optimizer's I/O cost estimate for a query exceeds the limit value. Such limits prevent the execution of potentially expensive queries.
- set statistics io on displays the number of actual logical and physical reads required. Use this information to set execution time resource limits. An execution time resource limit violation occurs when the actual I/O cost for a query exceeds the limit value.

Statistics for actual I/O cost include access costs only for user tables and worktables involved in the query. Adaptive Server may use other tables internally; for example, it accesses sysmessages to print out statistics. Therefore, there may be instances when a query exceeds its actual I/O cost limit, even though the statistics indicate otherwise.

In costing a query, the optimizer assumes that every page needed will require a physical I/O for the first access and will be found in the cache for repeated accesses. Actual I/O costs may differ from the optimizer's estimated costs, for several reasons.

The estimated cost will be higher than the actual cost if some pages are already in the cache or if the statistics are incorrect. The estimated cost may be lower than the actual cost if the optimizer chooses 16K I/O, and some of the pages are in 2K cache pools, which requires many 2K I/Os. Also, if a big join forces the cache to flush its pages back to disk, repeated access may require repeated physical I/Os.

The optimizer's estimates will not be accurate if the distribution or density statistics are out of date or cannot be used.

Calculating the I/O cost of a cursor

The cost estimate for processing a cursor is calculated at declare cursor time for all cursors except execute cursors, which is calculated when the cursor opens.

Pre-execution time resource limits on I/O cost are enforced at open *cursorname* time for all cursor types. The optimizer recalculates the limit value each time the user attempts to open the cursor.

An execution time resource limit applies to the cumulative I/O cost of a cursor from the time the cursor opens to the time it closes. The optimizer recalculates the I/O limit each time a cursor opens.

For a discussion of cursors, see Chapter 17, "Cursors: Accessing Data Row by Row," in the *Transact-SQL User's Guide*.

The scope of the *io_cost* limit type

A resource limit that restricts I/O cost applies only to single queries. If you issue several statements in a query batch, Adaptive Server evaluates the I/O usage for each query. For more information, see "Determining the scope of resource limits" on page 243.

Limiting elapsed time

Elapsed time is the number of seconds, in wall-clock time, required to execute a query batch or transaction. Elapsed time is determined by such factors as query complexity, server load, and waiting for locks.

To help develop appropriate limits for elapsed time use information you have gathered with set statistics time You can limit the elapsed time resource only at execution time.

With set statistics time set on, run some typical queries to determine processing time in milliseconds. Convert milliseconds to seconds when you create the resource limit.

Elapsed time resource limits are applied to all SQL statements in the limit's scope (query batch or transaction), not just to the DML statements. A resource limit violation occurs when the elapsed time for the appropriate scope exceeds the limit value.

Because elapsed time is limited only at execution time, an individual query will continue to run, even if its elapsed time exceeds the limit. If there are multiple statements in a batch, an elapsed time limit takes effect after a statement violates the limit and before the next statement is executed. If there is only one statement in a batch, setting an elapsed time limit has no effect.

Separate elapsed time limits are not applied to nested stored procedures or transactions. In other words, if one transaction is nested within another, the elapsed time limit applies to the outer transaction, which encompasses the elapsed time of the inner transaction. Therefore, if you are counting the wall-clock running time of a transaction, that running time includes all nested transactions.

The scope of the *elapsed_time* limit type

The scope of a resource limit that restricts elapsed time is either a query batch or transaction. You cannot restrict the elapsed time of a single query. For more information, see "Determining the scope of resource limits" on page 243.

Limiting the size of the result set

The row_count limit type limits the number of rows returned to the user. A limit violation occurs when the number of rows returned by a select statement exceeds the limit value.

If the resource limit issues a warning as its action, and a query exceeds the row limit, the full number of rows are returned, followed by a warning that indicates the limit value; for example:

```
Row count exceeded limit of 50.
```

If the resource limit's action aborts the query batch or transaction or kills the session, and a query exceeds the row limit, only the limited number of rows are returned and the query batch, transaction, or session aborts. Adaptive Server displays a message like the following:

```
Row count exceeded limit of 50.
Transaction has been aborted.
```

The row_count limit type applies to all select statements at execution time. You cannot limit an estimated number of rows returned at pre-execution time.

Determining row count limits

Use the *@@rowcount* global variable to help develop appropriate limits for row count. Selecting this variable after running a typical query can tell you how many rows the query returned.

Applying row count limits to a cursor

A row count limit applies to the cumulative number of rows that are returned through a cursor from the time the cursor opens to the time it closes. The optimizer recalculates the row_count limit each time a cursor opens.

The scope of the *row_count* limit type

A resource limit that restricts row count applies only to single queries, not to cumulative rows returned by a query batch or transaction. For more information, see "Determining the scope of resource limits" on page 243.

Creating a resource limit

Create a new resource limit with sp_add_resource_limit. The syntax is:

sp_add_resource_limit *name*, *appname*, *rangename, limittype*, *limit_value*, *enforced*, *action*, *scope*

Use this system procedure's parameters to:

- Specify the name of the user or application to which the resource limit applies.

  You must specify either a *name* or an *appname* or both. If you specify a user, the name must exist in the syslogins table. Specify "null" to create a limit that applies to all users or all applications.

- Specify the time range.

  The time range must already exist when you create the limit. For more information, see "Defining time ranges" on page 235.

- Specify the type of limit (io_cost, row_count, or elapsed_time), and set an appropriate value for the limit type.

  For more information, see "Choosing a limit type" on page 242.

- Specify whether the resource limit is enforced prior to or during query execution.

Specify numeric values for this parameter. Pre-execution time resource limits, which are specified as 1, are valid only for the io_cost limit. Execution time resource limits, which are specified as 2, are valid for all three limit types. For more information, see "Determining time of enforcement" on page 243.

- Specify the action to be taken (issue a warning, abort the query batch, abort the transaction, or kill the session).

  Specify numeric values for this parameter.

- Specify the scope (query, query batch, or transaction).

  Specify numeric values for this parameter. For more information, see "Determining the scope of resource limits" on page 243.

For detailed information, see sp_add_resource_limit in the *Adaptive Server Reference Manual*.

Resource limit examples

This section includes three examples of setting resource limits.

Example 1

```
sp_add_resource_limit NULL, payroll, tu_wed_7_10,
elapsed_time, 120, 2, 1, 2
```

This example creates a resource limit that applies to all users of the payroll application because the name parameter is NULL. The limit is valid during the tu_wed_7_10 time range. The limit type, elapsed_time, is set to a value of 120 seconds. Because elapsed_time is enforced only at execution time, the *enforced* parameter is set to 2. The *action* parameter is set to 1, which issues a warning. The limit's *scope* is set to 2, query batch, by the last parameter. Therefore, when the elapsed time of the query batch takes more than 120 seconds to execute, Adaptive Server issues a warning.

Example 2

```
sp_add_resource_limit joe_user, NULL,
saturday_night, row_count, 5000, 2, 3, 1
```

This example creates a resource limit that applies to all ad hoc queries and applications run by "joe_user" during the saturday_night time range. If a query (*scope* = 1) returns more than 5000 rows, Adaptive Server aborts the transaction (*action* = 3). This resource limit is enforced at execution time (*enforced* = 2).

Example 3

```
sp_add_resource_limit joe_user, NULL, "at all
times", io_cost, 650, 1, 3, 1
```

This example also creates a resource limit that applies to all ad hoc queries and applications run by "joe_user." However, this resource limit specifies the default time range, "at all times." When the optimizer estimates that the io_cost of the query (*scope* = 1) would exceed the specified value of 650, Adaptive Server aborts the transaction (*action* = 3). This resource limit is enforced at pre-execution time (*enforced* = 1).

Getting information on existing limits

Use sp_help_resource_limit to get information about existing resource limits.

Users who do not have the System Administrator role can use sp_help_resource_limit to list their own resource limits (only).

Users either specify their own login names as a parameter or specify the name parameter as "null." The following examples return all resource limits for user "joe_user" when executed by joe_user:

```
sp_help_resource_limit
```

or

```
sp_help_resource_limit joe_user
```

System Administrators can use sp_help_resource_limit to get the following information:

- All limits as stored in sysresourcelimits (all parameters NULL); for example:

  ```
  sp_help_resource_limit
  ```

- All limits for a given login (*name* is not NULL, all other parameters are NULL); for example:

  ```
  sp_help_resource_limit joe_user
  ```

- All limits for a given application (appname is not NULL; all other parameters are NULL); for example:

  ```
  sp_help_resource_limit NULL, payroll
  ```

- All limits in effect at a given time or day (either *limittime* or *limitday* is not NULL; all other parameters NULL); for example:

  ```
  sp_help_resource_limit @limitday = wednesday
  ```

- Limit, if any, in effect at a given time for a given login (*name* is not NULL, either *limittime* or *limitday* is not NULL); for example:

  ```
  sp_help_resource_limit joe_user, NULL, NULL,
  wednesday
  ```

For detailed information, see sp_help_resource_limit in the *Adaptive Server Reference Manual*.

Example of listing All existing resource limits

When you use sp_help_resource_limit without any parameters, Adaptive Server lists all resource limits within the server. For example:

```
                    sp_help_resource_limit
name     appname rangename rangeid limitid limitvalue enforced  action scope
----     ------- --------- ------- ------- ---------- --------  ------ -----
NULL     acctng  evenings        4       2        120         2       1     2
stein    NULL    weekends        1       3       5000         2       1     1
joe_user acctng  bus_hours       5       3       2500         2       2     1
joe_user finance bus_hours       5       2        160         2       1     6
wong     NULL    mornings        2       3       2000         2       1     1
wong     acctng  bus_hours       5       1         75         1       3     1
```

In the output, the rangeid column prints the value from systimeranges.id that corresponds to the name in the rangename column. The limitvalue column reports the value set by sp_add_resource_limit or sp_modify_resource_limit. Table 6-2 shows the meaning of the values in the limitid, enforced, action, and scope columns.

Table 6-2: Values for sp_help_resource_limit output

Column	Meaning	Value
limitid	What kind of limit is it?	1 I/O cost 2 Elapsed time 3 Row count
enforced	When is the limit enforced?	1 Before execution 2 During execution 3 Both
action	What action is taken when the limit is hit?	1 Issue a warning 2 Abort the query batch 3 Abort the transaction 4 Kill the session
scope	What is the scope of the limit?	1 Query 2 Query batch 4 Transaction 6 Query batch + transaction

If a System Administrator specifies a login name when executing sp_help_resource_limit, Adaptive Server lists all resource limits for that login. The output displays not only resource limits specific to the named user, but all resource limits that pertain to all users of specified applications, because the named user is included among all users.

For example, the following output shows all resource limits that apply to "joe_user". Because a resource limit is defined for all users of the acctng application, this limit is included in the output.

```
                sp_help_resource_limit joe_user
name     appname rangename rangeid limitid limitvalue enforced  action scope
----     ------- --------- ------- ------- ---------- --------  ------ -----
NULL     acctng  evenings        4       2        120         2       1     2
joe_user acctng  bus_hours       5       3       2500         2       2     1
joe_user finance bus_hours       5       2        160         2       1     6
```

Modifying resource limits

Use sp_modify_resource_limit to specify a new limit value or a new action to take when the limit is exceeded or both. You cannot change the login or application to which a limit applies or specify a new time range, limit type, enforcement time, or scope.

The syntax of sp_modify_resource_limit is:

sp_modify_resource_limit *name*, *appname*, *rangename*, *limittype*, *limitvalue*, *enforced*, *action*, *scope*

To modify a resource limit, specify the following values:

- You must specify a non-null value for either *name* or *appname*.
  - To modify a limit that applies to all users of a particular application, specify a *name* of "null."
  - To modify a limit that applies to all applications used by *name*, specify an *appnam*e of "null."
  - To modify a limit that governs a particular application, specify the application name that the client program passes to the Adaptive Server in the login packet.
- You must specify non-null values for *rangename* and *limittype*. If necessary to uniquely identify the limit, specify non-null values for *action* and *scope*.
- Specifying "null" for *limitvalue* or *action* indicates that its value does not change.

For detailed information, see sp_modify_resource_limit in the *Adaptive Server Reference Manual*.

Examples of modifying a resource limit

```
sp_modify_resource_limit NULL, payroll,
tu_wed_7_10, elapsed_time, 90, null, null, 2
```

This example changes the value of the resource limit that restricts elapsed time to all users of the *payroll* application during the tu_wed_7_10 time range. The limit value for elapsed time decreases to 90 seconds (from 120 seconds). The values for time of execution, action taken, and scope remain unchanged.

```
sp_modify_resource_limit joe_user, NULL,
saturday_night, row_count, NULL, NULL, 2, NULL
```

This example changes the action taken by the resource limit that restricts the row count of all ad hoc queries and applications run by "joe_user" during the saturday_night time range. The previous value for action was 3, which aborts the transaction when a query exceeds the specified row count. The new value is to 2, which aborts the query batch. The values for limit type, time of execution, and scope remain unchanged.

Dropping resource limits

Use sp_drop_resource_limit to drop a resource limit from an Adaptive Server.

The syntax is:

sp_drop_resource_limit {*name* , *appname* } [, *rangename*, *limittype*, *enforced*, *action*, *scope*]

Specify enough information to uniquely identify the limit. You must specify a non-null value for either *name* or *appname*. In addition, specify values according to those shown in Table 6-3.

Table 6-3: Identifying resource limits to drop

Parameter	Value specified	Consequence
name	• Specified login	Drops limits that apply to the particular login.
	• NULL	Drops limits that apply to all users of a particular application.
appname	• Specified application	Drops limits that apply to a particular application.
	• NULL	Drops limits that apply to all applications used by the specified login.
timerange	• An existing time range stored in the systimeranges system table	Drops limits that apply to a particular time range.
	• NULL	Drops all resource limits for the specified *name*, *appname*, *limittype*, enforcement time, *action*, and *scop*e, without regard to *rangename.*
limittype	• One of the three limit types: row_count, elapsed_time, io_cost	Drops limits that apply to a particular limit type.
	• NULL	Drops all resource limits for the specified *name*, *appname*, *timerange*, *action*, and scope, without regard to *limittype.*

Parameter	Value specified	Consequence
enforced	• One of the enforcement times: pre-execution or execution	Drops the limits that apply to the specified enforcement time.
	• NULL	Drops all resource limits for the specified *name*, *appname*, *limittype*, *timerange*, *action*, and scope, without regard to enforcement time.
action	• One of the four action types: issue warning, abort query batch, abort transaction, kill session	Drops the limits that apply to a particular action type.
	• NULL	Drops all resource limits for the specified *name*, *appname*, *timerange*, *limittype*, enforcement time, and *scop*e, without regard to *action.*
scope	• One of the scope types: query, query batch, transaction	Drops the limits that apply to a particular scope.
	• NULL	Drops all resource limits for the specified *name*, *appname*, *timerange*, *limittype*, enforcement time, and *action*, without regard to *scope.*

When you use sp_droplogin to drop an Adaptive Server login, all resource limits associated with that login are also dropped.

For detailed information, see sp_drop_resource_limit in the *Adaptive Server Reference Manual*.

Examples of dropping a resource limit

```
sp_drop_resource_limit NULL, payroll, tu_wed_7_10
```

This example drops all resource limits for all users of the payroll application during the tu_wed_7_10 time range.

```
sp_drop_resource_limit NULL, payroll, tu_wed_7_10, elapsed_time
```

This example is similar to the preceding example, but drops only the resource limit that governs elapsed time for all users of the payroll application during the tu_wed_7_10 time range.

```
sp_drop_resource_limit joe_user, payroll
```

This example drops all resource limits for “joe_user” from the payroll application.

Resource limit precedence

Adaptive Server provides precedence rules for time ranges and resource limits.

Time ranges

For each login session during the currently active time ranges, only one limit can be active for each distinct combination of limit type, enforcement time, and scope. The precedence rules for determining the active limit are as follows:

- If no limit is defined for the login ID for either the "at all times" range or the currently active time ranges, there is no active limit.
- If limits are defined for the login for both the "at all times" and time-specific ranges, then the limit for the time-specific range takes precedence.

Resource limits

Since either the user's login name or the application name, or both, are used to identify a resource limit, Adaptive Server observes a predefined search precedence while scanning the sysresourcelimits table for applicable limits for a login session. The following table describes the precedence of matching ordered pairs of login name and application name:

Level	Login name	Application name
1	joe_user	payroll
2	NULL	payroll
3	joe_user	NULL

If one or more matches are found for a given precedence level, no further levels are searched. This prevents conflicts regarding similar limits for different login/application combinations.

If no match is found at any level, no limit is imposed on the session.

CHAPTER 7

Configuring Character Sets, Sort Orders, and Languages

This chapter discusses Adaptive Server Enterprise internationalization and localization support issues.

Topics covered in this chapter include:

Understanding internationalization and localization

Internationalization is the process of enabling an application to support multiple languages and cultural conventions.

An internationalized application uses external files to provide language-specific information at execution time. Because it contains no language-specific code, an internationalized application can be deployed in any native language environment without code changes. A single version of a software product can be adapted to different languages or regions, conforming to local requirements and customs without engineering changes. This approach to software development saves significant time and money over the lifetime of an application.

Localization is the process of adapting an internationalized product to meet the requirements of one particular language or region, for example Spanish, including providing translated system messages; translations for the user interface; and the correct formats for date, time, and currency. One version of a softwared product may have many localized versions.

Sybase provides both internationalization and localization support. Adaptive Server includes the character set definition files and sort order definition files required for data processing support for the major business languages in Western Europe, Eastern Europe, the Middle East, Latin America, and Asia.

Sybase Language Modules provide translated system messages and formats for Chinese (Simplified), French, German, Japanese, Korean, Brazilian Portuguese, and Spanish. By default, Adaptive Server comes with U.S. English message files.

This chapter describes the available character sets and language modules and summarizes the steps needed to change the default character set, sort order, or message language for Adaptive Server.

Advantages of internationalized systems

The task of designing an application to work outside its country of origin can seem daunting. Often, programmers think that internationalizing means hard-coding dependencies based on cultural and linguistic conventions for just one country.

A better approach is to write an internationalized application: that is, one that examines the local computing environment to determine what language to use and loads files containing language-specific information at runtime.

When you use an internationalized application, a single application can be deployed in all countries. This has several advantages:

- You write and maintain one application, not half a dozen (or more).
- The application can be deployed, without change, in new countries as needed. You need only supply the correct localization files.
- All sites can expect standard features and behavior.

A sample internationalized system

An internationalized system may include internationalized client applications, gateways, and servers running on different platforms in different native language environments.

For example, an international system might include the following components:

- Order processing applications in New York City, Mexico City, and Paris (Client-Library applications)
- An inventory control server in Germany (Adaptive Server)
- An order fulfillment server in France (Adaptive Server)
- A central accounting application in Japan (an Open Server application working with an Adaptive Server)

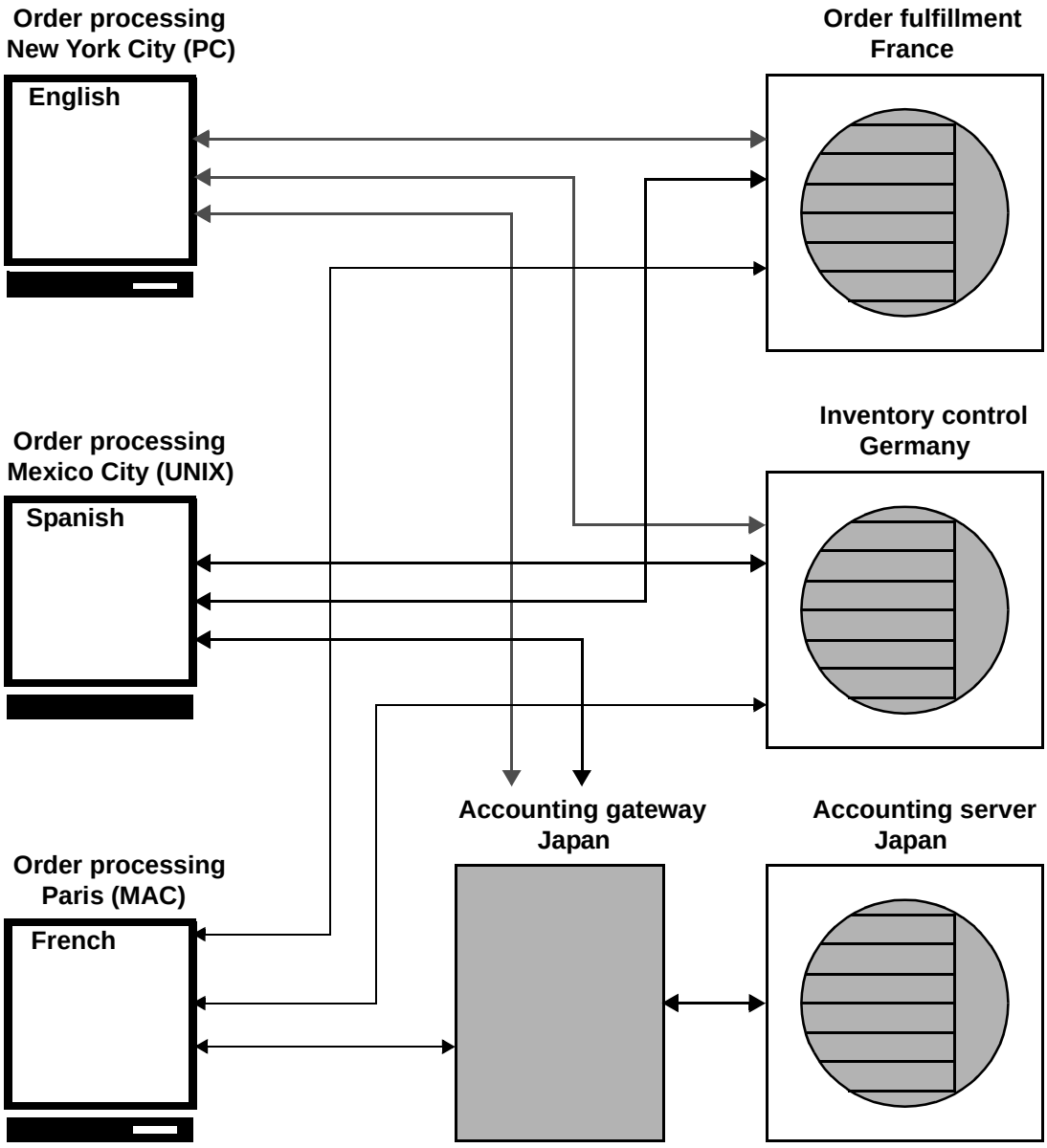
In this system, the order processing applications:

- Query the inventory control server to determine if requested items are in stock
- Place orders with the order fulfillment server
- Send financial information to the accounting application

The inventory control server and the order fulfillment server respond to queries, and the accounting application collects financial data and generates reports.

The system looks like this:

Figure 7-1: Example of an international system



In this example, all applications and servers use local languages and character sets to accept input and output messages.

Elements of an internationalized system

There are three elements that you can manipulate to configure your server language in an internationalized environment. Sybase suggests that you review these three elements and carefully plan the client/server network you want to create.

- Character set – the language in which the server sends and receives data to and from the client servers. Select the character set after carefully planning and analyzing the language needs of all client servers.
- Sort order – sort order options are dependent on the language and character set you select.
- System messages – messages display in one of several languages provided by Sybase. If your server language is not one of the languages provided, your system messages display in English, the default.

The following sections provide details about each of these elements.

Selecting the character set for your server

All data is encoded in your server in a special code. For example, the letter "a" is encoded as "97" in decimal. A **character set** is a specific collection of characters (including alphabetic and numeric characters, symbols, and nonprinting control characters) and their assigned numerical values, or codes. A character set generally contains the characters for an alphabet, for example, the Latin alphabet used in the English language, or a script such as Cyrillic used with languages such as Russian, Serbian, and Bulgarian. Character sets that are platform-specific and support a subset of languages, for example, the Western European languages, are called **native** or **national character sets**. All character sets that come with Adaptive Server, except for Unicode UTF-8, are native character sets.

A **script** is a writing system, a collection of all the elements that characterize the written form of a human language—for example, Latin, Japanese, or Arabic. Depending on the languages supported by an alphabet or script, a character set can support one or more languages. For example, the Latin alphabet supports the languages of Western Europe (see Group 1 in Table 7-1). On the other hand, the Japanese script supports only one language, Japanese. Therefore, the Group 1 character sets support multiple languages, while many character sets, such as those in Group 101, support only one language.

The language or languages that are covered by a character set is called a **language group.** A language group can contain many languages or only one language; a native character set is the platform-specific encoding of the characters for the language or languages of a particular language group.

Within a client/server network, you can support data processing in multiple languages *if all the languages belong to the same language group* (see Table 7-1). For example, if data in the server is encoded in a Group 1 character set, you could have French, German, and Italian data and any of the other Group 1 languages in the same database. However, you cannot store data from another language group in the same database. For example, you cannot store Japanese data with French or German data.

Unlike the native character sets just described, **Unicode** is an international character set that supports over 650 of the world's languages, such as Japanese, Chinese, Russian, French, and German. Unicode allows you to mix different languages from different language groups in the same server, no matter what the platform.

Since all character sets support the Latin script, and therefore English, a character set always supports at least two languages—English and one other language.

Many languages are supported by more than one character set. The character set you install for a language depends on the client's platform and operating system.

Adaptive Server supports the following languages and character sets:

Table 7-1: Supported languages and character sets

Language group	Languages	Character sets
Group 1	*Western European:* Albanian, Catalan, Danish, Dutch, English, Faeroese, Finnish, French, Galician, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish, Swedish	ASCII 8, CP 437, CP 850, CP 860, CP 863, CP 1252[a] , ISO 8859-1, ISO 8859-15, Macintosh Roman, ROMAN8
Group 2	*Eastern European:* Croatian, Czech, Estonian, Hungarian, Latvian, Lithuanian, Polish, Romanian, Slovak, Slovene (and English)	CP 852, CP 1250, ISO 8859-2, Macintosh Central European
Group 4	Baltic (and English)	CP 1257
Group 5	*Cyrillic:* Bulgarian, Byelorussian, Macedonian, Russian, Serbian, Ukrainian (and English)	CP 855, CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic
Group 6	Arabic (and English)	CP 864, CP 1256, ISO 8859-6
Group 7	Greek (and English)	CP 869, CP 1253, GREEK8, ISO 8859-7, Macintosh Greek
Group 8	Hebrew (and English)	CP 1255, ISO 8859-8
Group 9	Turkish (and English)	CP 857, CP 1254, ISO 8859-9, Macintosh Turkish, TURKISH8
Group 101	Japanese (and English)	CP 932 DEC Kanji, EUC-JIS, Shift-JIS
Group 102	Simplified Chinese (PRC) (and English)	CP 936, EUC-GB
Group 103	Traditional Chinese (ROC) (and English)	Big 5, CP 950[b] , EUC-CNS
Group 104	Korean (and English)	EUC-KSC
Group 105	Thai (and English)	CP 874, TIS 620
Group 106	Vietnamese (and English)	CP 1258
Unicode	Over 650 languages	UTF-8

a. CP 1252 is identical to ISO 8859-1 except for the 0x80–0x9F code points which are mapped to characters in CP 1252.
b. CP 950 is identical to Big 5.

Note The English language is supported by all character sets because the first 128 (decimal) characters of any character set include the Latin alphabet (defined as “ASCll-7). The characters beyond the first 128 differ between character sets and are used to support the characters in different native languages. For example, code points 0-127 of CP 932 and CP 874 both support English and the Latin alphabet. However, code points 128-255 support Japanese characters in CP 932 and code points 128-255 support Thai characters in CP 874.

The following character sets support the European currency symbol, the "euro": CP 1252 (Western Europe); CP 1250 (Eastern Europe); CP 1251 (Cyrillic) ; CP 1256 (Arabic); CP 1253 (Greek); CP 1255 (Hebrew); CP 1254 (Turkish); CP 874 (Thai); and Unicode UTF-8.

To mix languages from different language groups you *must* use Unicode. If your server character set is Unicode, you can support more than 650 languages in a single server and mix languages from any language group.

Selecting the server default character set

When you configure your server, you are asked to specify a default character set for the server. The default character set is the character set in which the server stores and manipulates data. Each server can have only one default character set.

By default, the installation tool assumes that the native character set of the platform operating system is the server's default character set. However, you can select any character set supported by Adaptive Server as the default on your server (see Table 7-1).

For example, if you are installing the server on IBM RS/6000 running AIX, and you select one of the Western European languages to install, the installation tool assumes the default character set to be ISO 8859-1.

If you are installing a Unicode server, select UTF–8 as your default character set.

For non-Unicode servers, determine what platform most of your client systems use and use the character set for this platform as the default character set on the server.

This has two advantages:

- The number of unmappable characters between character sets is minimized.

  Since there is usually not a complete one-to-one mapping between the characters in two character sets, there is a potential for some data loss. This is usually minor because most nonconverted characters are special symbols that are not commonly used or are specific to a platform.

- This minimizes the character set conversion that is required.

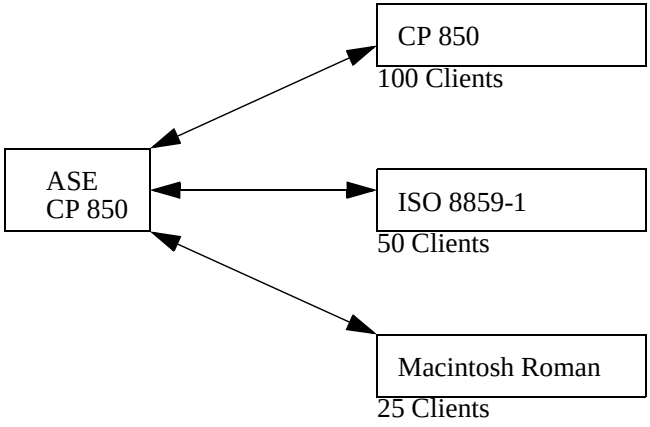
When the character set on the client system differs from the default character set on the server, data must be converted in order to ensure data integrity. Although the measured performance decrease that results from character set conversion is insignificant, it is good practice to select the default character set that results in the fewest conversions.

For example, if most of your clients use CP850, specify CP850 on your server. You can do this even if your server is on an HP-UX system (whose native character set for the Group 1 languages is ROMAN8).

Note Sybase strongly recommends that you decide which character set you want to use as your default before you create any databases or make any changes to the Sybase-supplied databases.

In the example below, 175 clients all access the same Adaptive Server. The clients are on different platforms and use different character sets. The critical factor that allows these clients to function together is that *all* of the character sets in the client/server system belong to the same language group (see Table 7-1). Notice that the default language for the Adaptive Server is CP 850, which is the character set used by the largest number of clients. This allows the server to operate most efficiently, with the least amount of character set conversion.

Figure 7-2: Clients using different character sets in the same language group



To help you choose the default character set for your server, the following tables list the most commonly used character sets by platform and language.

Table 7-2: Popular Western European client platforms

Platform	Language	Character set
Win 95, 98	U.S. English, Western Europe	CP 1252
Win NT 4.0	U.S. English, Western Europe	CP 1252
Win 2000	U.S. English, Western Europe	CP 1252
Sun Solaris	U.S. English, Western Europe	ISO 8859-1
HP-UX 10,11	U.S. Englsih, Western Europe	ISO 8859-1
IBM AIX 4.x	U.S. English, Western Europe	ISO 8859-1

Table 7-3: Popular Japanese client platforms

Platform	Language	Character set
Win 95, 98	Japanese	CP 932 for Windows
Win NT 4.0	Japanese	CP 932 for Windows
Win 2000	Japanese	CP 932 for Windows
Sun Solaris	Japanese	EUC-JIS
HP-UX 10,11	Japanese	EUC-JIS
IBM AIX 4.x	Japanese	EUC-JIS

Table 7-4: Popular Chinese client platforms

Platform	Language	Character set
Win 95, 98	Chinese (simplified)	CP 936 for Windows
Win NT 4.0	Chinese (simplified)	CP 936 for Windows
Win 2000	Chinese (simplified)	CP 936 for Windows
Sun Solaris	Chinese (simplified)	EUC-GB
HP-UX 10,11	Chinese (simplified)	EUC-GBS
IBM AIX 4.x	Chinese (simplified)	EUC-GB

Selecting the sort order

Different languages sort the same characters differently. For example, in English, *Cho* would be sorted before *Co*, whereas in Spanish, the opposite is true. In German, β is a single character, however in dictionaries it is treated as the double character *ss* and sorted accordingly. Accented characters are sorted in a particular order so that *aménité* comes before *amène*, whereas if you ignored the accents, the reverse would be true. Therefore, language-specific sort orders are required so that characters are sorted correctly.

Each character set comes with one or more sort orders that Adaptive Server uses to collate data. A sort order is tied to a particular language or set of languages and to a specific character set. The same sort orders can be used for English, French, and German because they sort the same characters identically, for example, *A*, *a*, *B*, *b*, and so on. Or the characters are specific to one of the languages—for example, the accented characters, *é* , *à*, and *á*, are used in French but not in English or German—and therefore, there is no conflict in how those characters are sorted. The same is not true for Spanish however, where the double letters *ch* and *ll* are sorted differently. Therefore, although the same character sets support all four languages, there is one set of sort orders for English, French and German, and a different set of sort orders for Spanish.

In addition, a sort order is tied to a particular character set. Therefore, there is one set of sort orders for English, French, and German in the ISO 8859-1 character set, another set in the CP 850 character set, and so on. The sort orders available for a particular character set are located in sort order definition files (**.srt* files) in the character set directory. For a list of character sets and their available sort orders, see "Available sort orders" on page 272.

Using sort orders

Sort orders are used to:

- Create Indexes
- Store data into indexed tables
- Specify an order by clause

Different types of sort orders

All character sets are offered with a binary sort order at a minimum, which blindly sorts all data based only on the arithmetic value of the code assigned to represent each letter (the “binary” code) in the character set. Binary sort order works well for the first 128 characters of each character set (ASCll English) and for Asian languages.When a character set supports more than one language (for example, Group 1 or Unicode) the binary sort order will most likely give incorrect results, and you should select another sort order.

Character sets may also have one or more of the dictionary sort orders listed below:

- *Dictionary order, case-sensitive, accent-sensitive*, sorts uppercase and lowercase letters separately. Dictionary order recognizes the various accented forms of a letter and sorts them after the associated unaccented letter.
- *Dictionary order, case-insensitive, accent-sensitive*, sorts data in dictionary order but does not recognize case differences. Uppercase letters are equivalent to their lowercase counterparts and are intermingled in sorting results. Useful for avoiding duplicate entries in tables of names.
- *Dictionary order, case-insensitive, accent-sensitive, order with preference*, does not recognize case difference in determining equivalency of items. A word in uppercase is equivalent to the same word in lowercase. Preference is given to uppercase letters (they appear first) if all other conditions are equal.

Using case-insensitive with preference may cause poor performance in large tables when the columns specified in an *order by* clause match the key of the table's clustered index. Do not select case-insensitive order with preference unless your installation requires that uppercase letters be sorted before lowercase letters in otherwise equivalent strings for *order by* clauses.

- *Dictionary order, case-insensitive, accent-insensitive*, treats accented forms of a letter as equivalent to the associated unaccented letter. It intermingles accented letters in sorting results.

Selecting the default sort order

Sybase servers can support only one default sort order at a time. If your users are using the same language or their languages use the same sort order, then select the desired sort order. For example, if your users are using French data and expect French sorting, then you can pick one of the French dictionary sort orders. Or if your users are using data in multiple languages and the languages use the same sort order, for example English, French, and German, you can pick one sort order and it will work for all your users in all languages.

However, if you have users using different languages that require different sort orders, for example French and Spanish, then you must select one of the sort orders as the default. If you pick, for example, a French sort order, your Spanish users will not see the *ch* and *ll* double characters sorted as they would expect. The installation procedure, by default, configures the server with the binary sort order.

You can use the sortkey function to setup customized alternative sort orders for your data—one for each language.These sort orders can be selected dynamically to meet the needs of different users . The sortkey function is separate from the default sort order, but can coexist in the same server. The range and depth of sort orders provided by the sortkey function is better than those provided by the default sort order mechanism. For more information, see sortkey and compare in the *Adaptive Server Reference Manual*.

Table 7-5: Available sort orders

Language or script	Character sets	Sort orders
All languages	UTF-8	Binary
Cyrillic: Bulgarian, Byelorussian, Macedonian, Russian, Serbian, Ukrainian	CP 855, CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic	Dictionary order, case sensitive, accent sensitive
English, French, German	ASCII 8, CP 437, CP850, CP 860, CP 863, CP 1252a, ISO 8859–1, ISO 8859–15, Macintosh Roman, ROMAN8	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, accent sensitive Dictionary order, case sensitive, accent sensitive, with preference Dictionary order, case insensitive, accent insensitive
English, French, German	CP 850	Alternate dictionary order, case sensitive Alternate dictionary order, case sensitive, accent insensitive Alternate dictionary order, case sensitive, with preference.
Greek	ISO 8859–7	Dictionary order, case sensitive, accent sensitive
Hungarian	ISO 8859–2	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, accent sensitive Dictionary order, case insensitive, accent insensitive
Russian	CP 866, CP 1251, ISO 8859–5, Koi8, Macintosh Cyrillic	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, accent sensitive
Scandinavian	CP 850	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, with preference
Spanish	ASCII 8, CP 437, CP850, CP 860, CP 863, CP 1252, ISO 8859–1, ISO 8859–15, Macintosh Roman, ROMAN8	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, accent sensitive Dictionary order, case insensitive, accent insensitive
Thai	CP 874, TIS 620	Dictionary order
Turkish	ISO 8859–9	Dictionary order, case sensitive, accent sensitive Dictionary order, case insensitive, accent insensitive Dictionary order, case insensitive, accent sensitive

If your language does not appear here, there is no language-specific sort order for your language. Select a binary sort order and then investigate whether the sortkey function meets your needs. As this table illustrates, many languages have more than one sort order.

Selecting the default Unicode sort order

The default Unicode sort order is distinctly different from the sort order for the server's default character set. This separate configuration parameter is a static parameter that requires that you restart your server and reindex the unichar data if it is changed. This sort order is identified using a string parameter, rather than a numeric parameter, to guarantee that the sort order is unique.

The available default Unicode sort orders are as follows:

Table 7-6: Default Unicode sort orders

Name	Description
default	default Unicode ML ordering
binary	default binary ordering
thaidict	Thai dictionary ordering
scandict	Scandinavian, dictionary
scannocp	Scandinavian, case insensitive
dict	English dictionary
nocase	English, case insensitive
noaccent	English, accent insensitive
espdict	Spanish, dictionary
espnocs	Spanish, case insensitive
espnoac	Spanish, accent insensitive
rusdict	Russian, dictionary
rusnocs	Russian, case insensitive
cyrdict	Cyrillic, dictionary
cyrnocs	Cyrillic, case insensitive
elldict	Greek, dictionary
hundict	Hungarian, dictionary
hunnoac	Hungarian, accent insensitive
hunnocs	hungarian, case insensitive
turkdict	Turkish, dictionary
turknoac	Turkish, accent insensitive
turknocs	Turkish, case insensitive
sjisbin	Japanese, sjis binary
iso14651	ISO 14651 standard ordering
eucjisbin	Japanese eucjis
gb2312bin	Chinese gb2312
cp932msbin	Japanese cp932
b165bin	Chinese b165
euckcsbin	Korean euckcs
utf8bin	matches Unicode UTF-8 binary sort order

You can add sort orders using external files in the *$SYBASE/collate/Unicode* directory. The names and collation IDs are stored in SYSCHARSETS. The names of external Unicode sort orders do not have to be in SYSCHARSETS before you can set the default Unicode sort order.

Note External Unicode sort orders are provided by Sybase. Do not attempt to create external Unicode sort orders.

Selecting a language for system messages

Any installation of Adaptive Server can use Language Modules containing files of messages in different languages. Adaptive Server provides Language Modules for messages in the following languages: English, Chinese (Simplified), French, German, Japanese, Korean, Brazilian Portuguese, and Spanish. If your client language is *not* one of these languages, you will see system messages in English, the default language.

Each client can choose to view messages in their own language at the same time, from the same server; for example, one client views system messages in French, another in Spanish, and another in German. To do this, however, all selected languages *must* be part of the same language group. For example, French, Spanish and German are all part of language group 1. Japanese, on the other hand, is part of language group 101, which contains no other languages. Therefore, if Japanese is your server language, you can display system messages only in Japanese or English. Remember that *all* language groups can display messages in English. There is also a server-wide default language, used if the user has not selected a specific language. If you use Unicode, you can view system messages in any of the supported languages.

You can select the language for your system messages in one of two ways:

- Select a language as part of your user profile.
- Enter a language in the *locales.dat* file.

The following table displays the supported system message languages and their language groups. Each user can select only one language for system messages per session.

Table 7-7: Supported system messages

Language group	System message languages	Character sets
Group 1	French, German, Spanish, Brazilian Portuguese	ASCII 8, CP 437, CP 850, CP 860, CP 863, CP 1252, ISO 8859-1, ISO 8859-15, Macintosh Roman, ROMAN8

Language group	System message languages	Character sets
Group 101	Japanese	CP 932, DEC Kanji, EUC-JIS, Shift-JIS
Group 102	Simplified Chinese (PRC)	CP 936, EUC-GB
Group 104	Korean	EUC-KSC
Unicode	French, German, Spanish, Brazilian Portuguese, Japanese, Simplified Chinese, Korean	UTF-8
All Other Language Groups	English	

You need to install language modules for all languages in which clients will receive messages. These language modules, located in the *locales* subdirectory of the Adaptive Server installation directory, are part of a group of files called *localization files*. For information about localization files and the software message directory structure, refer to "Types of localization files" on page 289.

Setting up your server: examples

This section discusses setup options and the steps necessary to implement them. This is only a sample, and is meant to suggest ideas and methods for your own setup process.

A Spanish-version server

This situation discusses how to set up a new server with all clients using the same language. To do this:

1 Select the server language, in this case, Spanish. By reviewing Table 7-1 on page 265, you see that Spanish is part of language group 1. Based on your platform, select a character set from language group 1. Sybase recommends that you select the character set used by the greatest number of clients. Or, if you think your company might someday expand into other countries and languages, you might consider installing Unicode (see "Selecting the character set for your server" on page 263).

2 Install the Spanish language module in the server. This allows clients to view system messages in Spanish.

3 Select the default sort order. By referring to Table 7-5 on page 272, you see that Spanish has three possible sort orders, in addition to binary sort order. Select a sort order.

4 Restart the server.

A U.S.-based company in Japan

This situation involves clients in Japan, who will want to enter data, sort data, and receive system messages in Japanese, while submitting data to a server that is accessed by English-only users. To do this:

1 Select the default character set for your server. If you install a character set from language group 101 (Japanese), you can support both Japanese and English data in the same server.

2 Install the Japanese language module so that system messages are available in Japanese.

3 Select the sort order. By referring to Table 7-5 on page 272, you can see that a binary sort order is the only sort order available for Japanese. Therefore, both the English and Japanese clients will have a default binary sort order. Consider using the sortkey function to provide solutions for both audiences.

4 Make sure that each Japanese user requests Japanese messages by default. Since you are using a character set from language group 101, and you have already installed the Japanese language module, your client in Japan will see messages in Japanese, while clients in the U.S. can choose to see messages in English.

A Japan-based company with multinational clients

This company is located in Japan, and has clients in France, Germany, and Spain.

This situation indicates that you will need to mix European and Asian languages in the same server.

1. Select the default server language and character set. Since your company is based in Japan and most of your clients are located in Japan, the default server language should be Japanese. But you also want your clients in France, Germany, and Spain to be able to send and receive data in their native languages. By reviewing Table 7-1 on page 265, you can see that Japanese is part of language group 101, while French, German, and Spanish are part of language group 1. Since the languages you need are not part of the same language group, the only way you can have all of these languages on the same server is to select Unicode as your default character set.
2. Install the language modules for Japanese, French, German, and Spanish.
3. Select the binary sort order, since this is the only sort order available for the Unicode character set. (You can, however, consider using the sortkey function inside your application code to supply data sorted according to each user's preference.)
4. Select Japanese as the default language for system messages. Clients in other countries can select their own native language for messages.

Changing the character set, sort order, or message language

Even after you have configured your server, a System Administrator can change the default character set, sort order, or message language used by Adaptive Server. Because a sort order is built on a specific character set, changing character sets always involves a change in sort order. However, you can change the sort order without changing character sets, because more than one sort order may be available for a character set.

To display Adaptive Server's default sort order, character set, and a table of its primary sort orders, enter:

```
sp_helpsort
```

Changing the default character set

Adaptive Server can have only one *default character set*, the character set in which data is stored in its databases. When you install Adaptive Server, you specify a default character set.

Warning! Please read the following carefully and exercise caution when changing the default character set in Adaptive Server. Sybase strongly recommends that you perform backups before you change a default character set.

When you change the default character set in Adaptive Server, you need to convert any existing data to the new default character set. Conversion is unnecessary *only* if:

- There is no user data in the server.
- It is acceptable to destroy user data in the server.
- You are *absolutely certain* that data in the server uses only ASCll-7. In this case, you can change the default without first copying your data out of the server.

In all other cases, you must convert the existing data as follows:

1 Copy the data out using bcp.

2 Change the default character set.

3 Use bcp with the appropriate flags for data conversion to copy the data back into the server.

See the *Utility Guide* for more information about using bcp to copy data.

Warning! After converting data to a different character set (particularly to UTF-8), the data may be too large for the allocated column size. Re-create the columns affected with a larger size. Refer to the Unidb tool in the Sybase UDK product.

Code conversion between the character set of the existing data and the new default character set must be supported. If it is not, conversion errors will occur and the data will not be converted correctly. See Chapter 8, "Configuring Client/Server Character Set Conversions," for more information about supported character set conversions.

Even if conversions are supported between the character sets, some errors may occur due to minor differences between the character sets, or because some characters do not have equivalents in other character sets. Rows containing problematic data may not get copied back into the database or data may contain partial or invalid characters.

Changing the default sort order

Adaptive Server can have only one *default sort order*, the collating sequence it uses to order data. When you consider changing the sort order for character data on a particular Adaptive Server, keep this in mind: all of your organization's Adaptive Servers should have the same sort order. A single sort order enforces consistency and makes distributed processing easier to administer.

You may have to rebuild your indexes after changing the default sort order. For more information, see "Reconfiguring the character set, sort order, or message language" on page 280.

Reconfiguring the character set, sort order, or message language

This section summarizes the steps to take before and after changing Adaptive Server's default character set, sort order, or message language. For procedures on how to configure the character set, sort order, or message language for a new server, see the configuration documentation for your platform.

If your data does not have to be converted to a new character set and both the old and the new character sets use binary sort order you can use a databse dump. You can restore your database from backups that were made before the character set was reconfigured.

Note Back up all databases in Adaptive Server both before and after you change character sets or sort orders.

Usually, you cannot reload your data from a database dump when you have reconfigured the default character set and sort order.

If the following is true, use bcp to copy the data out of and into your databases.

- If a database contains character data, and you want the data to be converted to a new character set. Do not load a database dump of the data into an Adaptive Server with the new default character set. Adaptive Server interprets the data loaded as if it is in the new character set, and the data will be corrupted.
- If you are changing only the default sort order and not the default character set. You cannot load a database from a dump that was performed before you changed the sort order. If you attempt to do so, an error message appears, and the load is aborted.
- You change the default character set, and either the old or the new sort order is not binary. You cannot load a database dump that was made before you changed the character set.

Preliminary steps

Before you run the installation program to reconfigure Adaptive Server:

1 Dump all user databases and the master database. If you have made changes to model or sybsystemprocs, dump them also.

2 Load the Language Module if it is not already loaded (see the configuration documentation for your platform for complete instructions).

3 If you are changing the Adaptive Server default character set, and your current databases contain non ASCII-7 data, use bcp to copy the existing data out of your databases.

Once you have loaded the Language Module, you can run the Adaptive Server installation program, which allows you to:

- Install or remove message languages and character sets included with Adaptive Server
- Change the default message language or character set
- Select a different sort order

See the configuration documentation for your platform for instructions on using the installation program.

To reconfigure the language, character set, or sort order, use the sqlloc utility, described in Utility Programs for UNIX Platforms. If you are using Windows NT, use the Server Config utility, described in Configuring Adaptive Server for Windows NT. If you are adding a new character set that is not included with Adaptive Server, see the *Sybase Character Sets* manual for complete instructions.

If you installed additional languages but did not change Adaptive Server's character set or sort order, you have completed the reconfiguration process.

If you changed the Adaptive Server default character set, and your current databases contain non ASCII-7 data, copy your data back into your databases, using bcp with the necessary flags to enable conversion.

If you changed Adaptive Server's default sort order or character set, see "Reconfiguring the character set, sort order, or message language" on page 280.

Setting the user's default language

If you install an additional language, users running client programs can run sp_modifylogin to set that language as their default language, or set the LANG variable on the client machine, with the appropriate entries in locales.dat.

Recovery after reconfiguration

Every time Adaptive Server is stopped and restarted, recovery is performed automatically on each database. Automatic recovery is discussed in detail in Chapter 26, "Developing a Backup and Recovery Plan."

After recovery is complete, the new sort order and character set definitions are loaded.

If you have changed the sort order, Adaptive Server switches to single-user mode to allow the necessary updates to system tables and to prevent other users from using the server. Each system table with a character-based index is automatically checked to see if any indexes have been corrupted by the sort order change. Character-based indexes in system tables are automatically rebuilt, if necessary, using the new sort order definition.

After the system indexes are rebuilt, character-based user indexes are marked "suspect" in the sysindexes system table, without being checked. User tables with suspect indexes are marked "read-only" in sysobjects to prevent updates to these tables and use of the "suspect" indexes until they have been checked and, if necessary, rebuilt.

Next, the new sort order information replaces the old information in the area of the disk that holds configuration information. Adaptive Server then shuts down so that it starts for the next session with a complete and accurate set of system information.

Using *sp_indsuspect* to find corrupt indexes

After Adaptive Server shuts down, restart it, and use sp_indsuspect to find the user tables that need to be reindexed. The syntax is:

```
sp_indsuspect [tab_name]
```

where *tab_name* is the optional name of a specific table. If *tab_name* is missing, sp_indsuspect creates a list of all tables in the current database that has indexes marked "suspect" when the sort order changes.

In this example, running sp_indsuspect in mydb database yields one suspect index:

```
sp_indsuspect
Suspect indexes in database mydb
Own.Tab.Ind (Obj_ID, Ind_ID) =
dbo.holdings.h_name_ix(160048003, 2)
```

Rebuilding indexes after changing the sort order

dbcc reindex checks the integrity of indexes on user tables by running a "fast" version of dbcc checktable. For details, see "dbcc checktable" on page 729. dbcc reindex drops and rebuilds the indexes where the sort order used is not consistent with the new sort order. When dbcc reindex discovers the first index-related error, it displays a message, and then rebuilds the inconsistent indexes. The System Administrator or table owner should run dbcc reindex after changing the sort order in Adaptive Server.

The syntax is:

```
dbcc reindex ({table_name | table_id})
```

Run this command on all tables listed by sp_indsuspect as containing suspect indexes. For example:

```
dbcc reindex(titles)
One or more indexes are corrupt. They will be
rebuilt.
```

In the preceding example, dbcc reindex discovers one or more suspect indexes in the table titles; it drops and re-creates the appropriate indexes.

If the indexes for a table are already correct, or if there are no indexes for the table, dbcc reindex does not rebuild any indexes. It displays a message instead. If a table is suspected of containing corrupt data, the command is aborted. If that happens, an error message instructs the user to run dbcc checktable.

When dbcc reindex finishes successfully, all "suspect" marks on the table's indexes are removed. The "read-only" mark on the table is also removed, and the table can be updated. These marks are removed whether or not any indexes have to be rebuilt.

dbcc reindex does not reindex system tables. System indexes are checked and rebuilt, if necessary, as an automatic part of recovery after Adaptive Server is restarted following a sort order change.

Upgrading *text* data after changing character sets

If you have changed an Adaptive Server's character set to a **multibyte character set** use dbcc fix_text to upgrade text values.

The syntax is:

dbcc fix_text ({*table_name* | *table_id*})

Changing to a multibyte character set makes the management of text data more complicated. A text value can be large enough to cover several pages; therefore, Adaptive Server must be able to handle characters that span page boundaries. To do so, Adaptive Server requires additional information on each of the text pages. The System Administrator or table owner must run dbcc fix_text on each table that has text data to calculate the new values needed.

To see the names of all tables that contain text data, use:

```
select sysobjects.name
from sysobjects, syscolumns
where syscolumns.type = 35
and sysobjects.id = syscolumns.id
```

The System Administrator or table owner must run dbcc fix_text to calculate the new values needed.

The syntax of dbcc fix_text is:

```
dbcc fix_text (table_name | table_id)
```

The table named must be in the current database.

dbcc fix_text opens the specified table, calculates the character statistics required for each text value, and adds the statistics to the appropriate page header fields. This process can take a long time, depending on the number and size of the text values in a table. dbcc fix_text can generate a large number of log records, which may fill up the transaction log. dbcc fix_text performs updates in a series of small transactions so that if a log becomes full, only a small amount of work is lost.

If you run out of log space, clear out your log (see Chapter 27, "Backing Up and Restoring User Databases"). Then restart dbcc fix_text, using the same table that was being upgraded when the original dbcc fix_text halted. Each multibyte text value contains information that indicates whether it has been upgraded, so dbcc fix_text upgrades only the text values that were not processed in earlier passes.

If your database stores its log on a separate segment, you can use thresholds to manage clearing the log. See Chapter 29, "Managing Free Space with Thresholds."

If dbcc fix_text cannot acquire a needed lock on a text page, it reports the problem and continues with the work, like this:

```
Unable to acquire an exclusive lock on text page 408.
This text value has not been recalculated.  In order
to recalculate those TEXT pages you must release the
lock and reissue the dbcc fix_text command.
```

Retrieving *text* values after changing character sets

If you attempt to retrieve text values after changing to a multibyte character set, and you have not run dbcc fix_text, the command fails with this error message:

```
Adaptive Server is now running a multi-byte
character set, and this TEXT column's character
counts have not been recalculated using this
character set. Use dbcc fix_text before running this
```

```
    query again.
```

Note If you have changed the sort order or character set and errors occurred, see "How to Manually Change Sort Order or Default Character Set" in the *Adaptive Server Enterprise Troubleshooting and Error Messages Guide*.

Installing date strings for unsupported languages

You can use sp_addlanguage to install names for the days of the week and months of the year for languages that do not have Language Modules. With sp_addlanguage, you define:

- A language name and (optionally) an alias for the name
- A list of the full names of months and a list of abbreviations for the month names
- A list of the full names of the days of the week
- The date format for entering dates (such as month/day/year)
- The number of the first day of the week

This example adds the information for Italian:

```
sp_addlanguage italian, italiano,
"gennaio,febbraio,marzo,aprile,maggio,giugno,luglio,agosto,settembre,ottobre,
novembre,dicembre",
"genn,feb,mar,apr,mag,giu,lug,ago,sett,ott,nov,dic",
"lunedi,martedi,mercoledi,giovedi,venerdi,sabato,domenica",
dmy, 1
```

sp_addlanguage enforces strict data entry rules. The lists of month names, month abbreviations, and days of the week must be comma-separated lists with no spaces or line feeds (returns). Also, they must contain the correct number of elements (12 for month strings, 7 for day-of-the-week strings.)

Valid values for the date formats are: mdy, dmy, ymd, ydm, myd, and dym. The dmy value indicates that the dates are in day/month/year order. This format affects only data entry; to change output format, you must use the convert function.

Server versus client date interpretation

Generally, date values are resolved on the client. When a user selects date values, Adaptive Server sends them to the client in internal format. The client uses the *common.loc* file and other localization files in the default language subdirectory of the *locales* directory on the client to convert the internal format to character data. For example, if the user's default language is Spanish, Adaptive Server looks for the *common.loc* file in */locales/spanish/char_set*. It uses the information in the file to display, for example, 12 febrero 1997.

Assume that the user's default language is set to Italian, a language for which Adaptive Server does not provide a Language Module, and that the date values in Italian have been added. When the client connects to the server and looks for the *common.loc* file for Italian, it does not find the file. The client prints an error message and connects to the server. If the user then selects date values, the dates are displayed in U.S. English format.To display the date values added with sp_addlanguage, use the convert function to force the dates to be converted to character data at the server.

The following query generates a result set with the dates in U.S. English format:

```
select pubdate from titles
```

whereas the query below returns the date with the month names in Italian:

```
select convert(char(19),pubdate) from titles
```

Internationalization and localization files

Types of internationalization files

The files that support data processing in a particular language are called *internationalization files*. Several types of internationalization files come with Adaptive Server. Table 7-8 describes these files.

Table 7-8: Internationalization files

File	Location	Purpose and contents
charset.loc	In each character set subdirectory of the *charsets* directory	Character set definition files that define the lexical properties of each character, such as alphanumeric, punctuation, operand, and uppercase or lowercase. Used by Adaptive Server to correctly process data.
**.srt*	In each character set subdirectory of the *charsets* directory	Defines the sort order for alphanumeric and special characters, including ligatures, diacritics, and other language-specific considerations.
**.xlt*	In each character set subdirectory of the *charsets* directory	Terminal-specific character translation files for use with utilities such as bcp and isql. For more information about how the.*xlt* files are used, see Chapter 8, "Configuring Client/Server Character Set Conversions," and the *Utility Guide*..

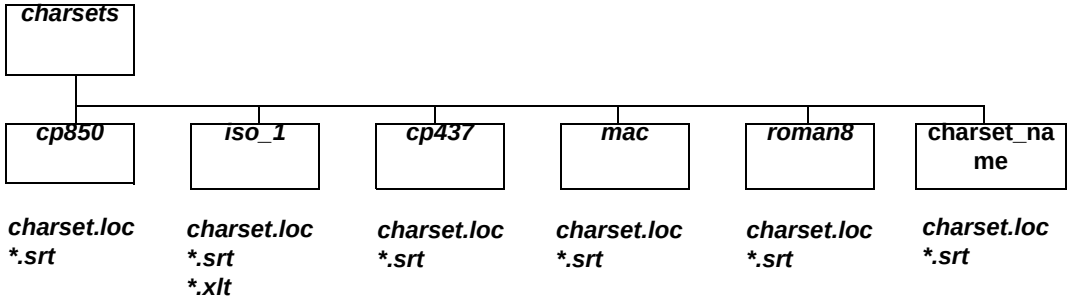
Warning! Do not alter any of the internationalization files. If you need to install a new terminal definition or sort order, contact your local Sybase office or distributor.

Character sets directory structure

Figure 7-3 shows the directory structure for the Western European character sets that come with Adaptive Server. There is a separate subdirectory for each character set in the *charsets* directory. Within the subdirectory for each character set (for example, *cp850*) are the character set and sort order definition files and terminal-specific files.

If you load additional character sets, they will also appear in the *charsets* directory :

Figure 7-3: Structure of the charsets directory



The following global variables contain information about character sets:

@@char_convert	Contains 0 if character set conversion is not in effect. Contains 1 if character set conversion is in effect.
@@client_csname	The client's character set name. Set to NULL if client character set has never been initialized; otherwise, it contains the name of the character set for the connection.
@@client_csid	The client's character set ID. Set to -1 if client character set has never been initialized; otherwise, it contains the client character set ID from syscharsets for the connection.
@@maxcharlen	The maximum length, in bytes, of a character in Adaptive Server's default character set.
@@ncharsize or *@@charsize?*	The maximum length, in bytes, of a character set in the current server default character set.
@@unicharsize	Equals 2.

Types of localization files

Adaptive Server includes several localization files for each Language Module, as shown in Table 7-9.

Table 7-9: Localization files

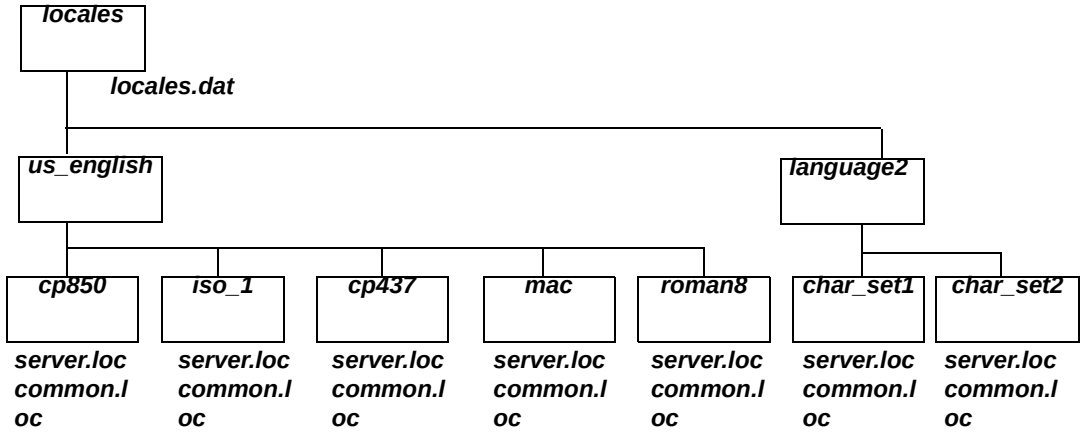
File	Location	Purpose and Contents
locales.dat	In the *locales* directory	Used by client applications to identify the default message language and character set.
server.loc	In the character set subdirectories under each language subdirectory in the *locales* directory	Software messages translated into the local language. Sybase products have product-specific **.loc* files. If an entry is not translated, that software message or string appears in U.S. English instead of the local language.
common.loc	In each language and character set directory of the *locales* directory	Contains the local names of the months of the year and their abbreviations and information about the local date, time, and money formats.

Warning! Do not alter any of the localization files. If you need to alter any information in those files, contact your local Sybase office or distributor.

Software messages directory structure

Figure 7-4 shows how localization files are arranged. Within the *locales* directory is a subdirectory for each language installed. There is always a *us_english* subdirectory. (On PC platforms, this directory is called *english*.) During installation, when you are prompted to select the languages you want installed on Adaptive Server, the install program lists the supported software message languages. If you install Language Modules for additional languages, you will see subdirectories for those languages. Within each language are subdirectories for the supported character sets; for example, *cp850* is a supported character set for *us_english*. Software message files for each Sybase product reside in the character set subdirectories.

Figure 7-4: Messages directory structure



Message languages and global variables

The following global variables contain information about languages:

@@langid	Contains the local language ID of the language currently in use (specified in syslanguages.langid)
@@language	Contains the name of the language currently in use (specified in syslanguages.name)

CHAPTER 8

Configuring Client/Server Character Set Conversions

This chapter describes how to configure character set conversion when the client uses a different character set than Adaptive Server.

Topics covered in this chapter include:

Character set conversion in Adaptive Server

In a heterogeneous environment, Adaptive Server may need to communicate with clients running on different platforms using different character sets. Although different character sets may support the same language group (for example, ISO 8858-1 and CP 850 support the group 1 languages), they may encode the same characters differently. For example, in ISO 8859-1, the character *à* is encoded as *0xE0* in hexadecimal. However, in CP 850 the same character is encoded as *0x85* in hexadecimal.

To maintain data integrity between your clients and servers, data must be converted between the character sets. The goal is to ensure that an “a” remains an “a” even when crossing between machine and character set boundaries. This process is known as *character set conversion.*

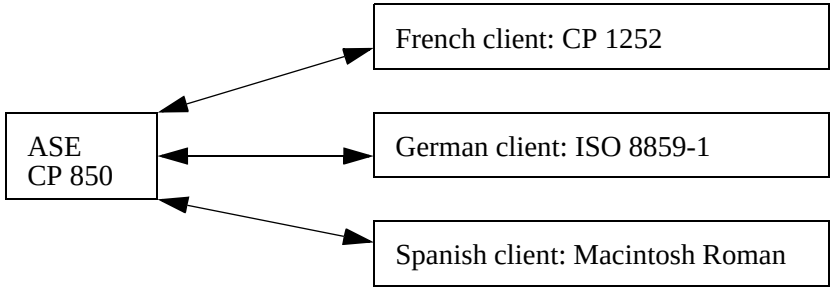
Supported character set conversions

Character set conversion occurs between a pair of character sets. The supported conversions in any particular client/server system depend on the character sets used by the server and its clients. One type of character set conversion occurs if the server uses a native character set as the default; a different type of conversion is used if the server default is Unicode UTF-8.

Conversion for native character sets

Adaptive Server supports character set conversion between native character sets belonging to the same language group. If the server has a native character set as its default, the clients' character sets must belong to the same language group. Figure 8-1is an example of a Western European client/server system. In this example, the clients' character sets and Adaptive Server's default character set all belong to Group 1. Data is correctly converted between the client character sets and the server default character set. Since they all belong to the same language group, the clients can view all data on the server, no matter which client submitted the data.

Figure 8-1: Character set conversion when server and client character sets belong to the same language group

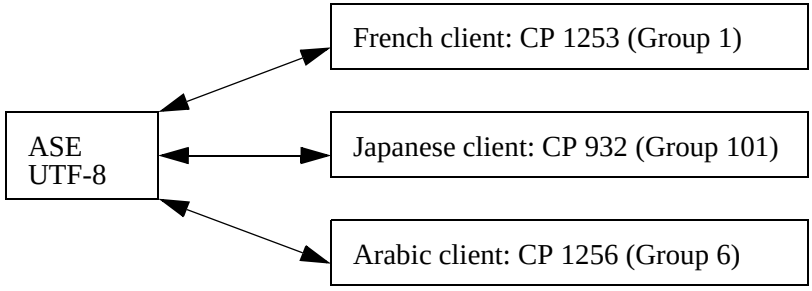


For a list of the language groups and supported character sets, see "Supported languages and character sets" on page 265.

Conversion in a Unicode system

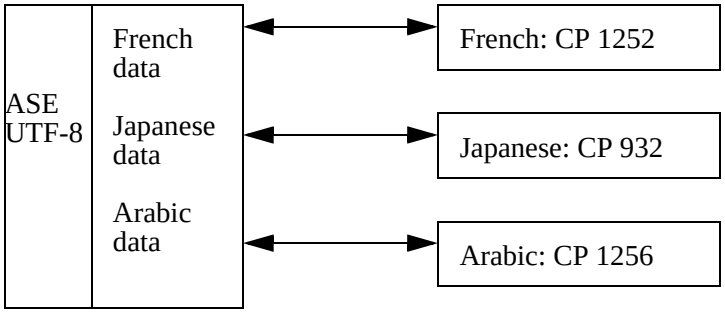
Adaptive Server also supports character set conversion between UTF-8 and any native character set that Sybase supports. In a Unicode system, since the server default character set is UTF-8, the client character set may be a native character set from any language group. Therefore, a Japanese client (group 101), a French client (Group 1), and an Arabic client (group 6) can all send and receive data from the same server. Data from each client is correctly converted as it passes between each client and the server.

Figure 8-2: Character set conversion in a Unicode system



Note however, that each client can view data only in the language supported by its character set. Therefore, the Japanese client can view any Japanese data on the server, but it cannot view Arabic or French data. Likewise, the French client can view French or any other Western European language supported by its character set, but not Japanese or Arabic.

Figure 8-3: Viewing Unicode data



An additional character set, ASCII 7, is a subset of *every* character set, including Unicode, and is therefore compatible with all character sets in all language groups. If either the Adaptive Server or the client's character set is ASCII 7, any 7-bit ASCII character can pass between the client and server unaltered and without conversion.

Sybase does not recommend that you configure a server for ASCII-7, but you can achieve the same benefits of compatibility by restricting each client to use only the first 128 characters of each native character set.

Types of character set conversion

Character set conversion is implemented on Adaptive Server in two different ways:

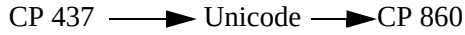
- Adaptive Server direct conversions
- Unicode conversions

Adaptive Server direct conversions

Adaptive Server direct conversions support conversions between two native character sets of the *same* language group. For example, Adaptive Server supports conversion between CP 437 and CP 850, because both belong to the group 1 language group. Adaptive Server direct conversions exist between many, but not all, native character sets of a language group (see Table 8-1 on page 299).

Unicode conversions

Unicode conversions exists for all native character sets. When converting between two native character sets, Unicode conversion uses Unicode as an intermediate character set. For example, to convert between the server default character set (CP 437), and the client character set (CP 860), CP 437 is first converted to Unicode; Unicode is then converted to CP 860.



As this example illustrates, Unicode conversions may be used either when the default character set of the server is UTF-8, or a native character set. You must specifically configure your server to use Unicode conversions (unless the server's default character set is UTF-8).

Earlier versions of Adaptive Server used direct conversions, and it is the default method for character set conversions. However, Unicode conversions implemented in more recent versions of Adaptive Server releases allow easier and less complex character set conversion. Sybase continues to support existing Adaptive Server direct conversions, but Sybase now also uses Unicode conversions to provide complete conversion support for all character sets. Sybase has no plans to add new direct conversions.

Which type of conversion do I use?

To determine the conversion options that are available for your client/server system, see Table 8-1 on page 299.

Non-Unicode client/server systems

In a non-Unicode system, the character sets of the server and clients are native character sets; therefore, you can use the Adaptive Server direct conversions.

However, there are some character sets for which there is no Adaptive Server direct conversion; in this situation, you must use Unicode conversions.

- If all character sets in your client/server system fall into column 1 of Table 8-1, use the Adaptive Server direct conversions. The character sets must all belong to the same language group.
- If the character sets in your client/server system fall into column 2 of Table 8-1, or some combination of columns 1 and 2, then you *must* configure your server to use Unicode conversions. Again, the character sets must all belong to the same language group.

For example, assume the server default character set is CP 850 and the clients' character sets are either ISO 8859-1 or ROMAN 8. Table 8-1 shows that direct conversions exist between CP 850 and the client character sets. Now, suppose you add a client using CP 1252 to this configuration. Since there is no direct conversion between CP 1252 and CP 850, (the default server character set), you *must* use Unicode conversions to convert between CP 1252 and CP 850. When you have a mixture of character sets—some where you can use Adaptive Server direct conversions and others where you must use Unicode conversions—you can specify that a combination of Adaptive Server direct conversion and Unicode conversion be used.

Unicode client/server systems

If your server default is Unicode UTF-8, then all conversions are between UTF-8 and whatever native character set is being used on the client systems. Therefore, in a Unicode system, Unicode conversions are used *exclusively*.

Table 8-1: Conversion methods for character sets

	Column 1	Column 2
Language group	**Adaptive Server direct conversions and Unicode conversions**	**Unicode conversions only**
Group 1	CP 437, CP 850, ISO 8859-1, Macintosh Roman, ROMAN8	CP 860, CP 1252, ISO 8859-15, CP 863
Group 2	CP 852, CP 1250, CP 8859-1, Macintosh Central European	ISO 8859-2
Group 4	No conversions needed (only one character set supported)	
Group 5	CP 855, CP 866, CP 1251, ISO 8859-5, Koi8, Macintosh Cyrillic	
Group 6		CP 864, CP 1256, ISO 8859-6
Group 7	CP 869, CP 1253, GREEK8, ISO 8859-7, Macintosh Greek	
Group 8		CP 1255, ISO 8859-8
Group 9	CP 857, CP 1254, ISO 8859-9, Macintosh Turkish, TURKISH8	
Group 101	DEC Kanjii, EUC-JIS, Shift-JIS	CP 932
Group 102		CP 936, EUG-GB
Group 103		Big 5, CP 950, EUC-CNS
Group 104	No conversions needed (only one character set supported)	
Group 105		CP 874, TIS 620
Group 106	No conversions needed (only one character set supported)	
Unicode	No conversions needed (only one character set supported)	

Configuring the server

By default, Adaptive Server uses direct conversions to convert data between different character sets. To use the Unicode conversions, you must configure the server with the sp_configure command. Set the enable unicode conversions option to either 1 or 2.

- If you set sp_configure "enable unicode conversions" to 1:

  This setting uses Adaptive Server direct conversions or Unicode conversions. Adaptive Server first checks to see if an Adaptive Server direct conversion exists for the server and client character set. If direct conversion, it uses the direct conversion exists, it is used; if no direct conversion exists, the Unicode conversion is used.

  Use this setting if the character sets in your client/server system fall into both columns 1 and 2 in Table 8-1.

- If you set sp_configure "enable unicode conversions" to 2:

  This setting uses Unicode conversions *only*. Adaptive Server uses Unicode conversions, without attempting to find an Adaptive Server direct conversion.

  Use this setting if the client/server conversions result in a change in the data length (see "Conversions and changes to data lengths" on page 302)

If all character sets fall into column 2 in Table 8-1, then you should set enable unicode conversions to 2 to always use Unicode conversions.

If the server default is UTF-8, the server automatically uses Unicode conversions only.

Enabling and disabling character set conversion

When a client requests a connection, the client identifies its character set to Adaptive Server. Adaptive Server compares the client character set with its default character set, and if the two names are identical, no conversion is required. If the names differ, Adaptive Server determines whether it supports conversion between its default and the client's character set. If it does not, it send an error message to the client and continues with the logon process. If it does, then character set conversion is automatically enabled. If the default character set of the server is UTF-8, it automatically uses Unicode conversions. If the default is a native character set, the server uses ASE direct conversions, unless the user specifies that Unicode conversions be used.

You can disable character set conversion at the server level. You may want to do this if:

- All of your clients are using the same character set as the server default, and therefore, no conversion is required.
- Conversion between the client character set and the server default is not supported.
- You want to store data in the server without converting the data, that is, without changing the encoding of the data.

To disable character set conversion at the server level, set the disable character set conversion parameter to 1. No conversion will occur for any client connecting to the server. By default this parameter is set to 0, which enables conversions.

You can also control character set conversion at the connection level using the set char_convert command from within a client session. set char_convert off turns conversion off between a particular client and the server. You may want to set char_convert off if the client and the server use the same character set, which makes conversion unnecessary. set char_convert on turns conversion back on.

Characters that cannot be converted

During the conversion process, some characters may not be converted. Here are two reasons:

- The character exists (is encoded) in the source character set, but it does not exist in the target character set. For example, the OE ligature, is part of the Macintosh character set (code point 0xCE). This character does not exist in the ISO 8859-1 character set. If the OE ligature exists in data that is being converted from the Macintosh to the ISO 8859-1 character set, it causes a conversion error.
- The character exists in both the source and the target character set, but in the target character set, the character is represented by a different number of bytes than in the source character set.

  For example, 1-byte accented characters (such as á, è) are 2-byte characters in UTF-8; 2-byte Thai characters are 3-byte characters in UTF-8. You can avoid this limitation by configuring the enable unicode conversion option to 1 or 2.

Error handling in character set conversion

Adaptive Server's character set conversion reports conversion errors when a character exists in the client's character set but not in the server's character set, or vice versa. Adaptive Server must guarantee that data successfully converted on input to the server can be successfully converted back to the client's character set when the client retrieves that data. To do this effectively, Adaptive Server must avoid putting suspect data into the database.

When Adaptive Server encounters a conversion error in the data being entered, it generates this message:

```
Msg 2402, Severity 16 (EX_USER):
Error converting client characters into server's
character set. Some character(s) could not be
converted.
```

A conversion error prevents query execution on insert and update statements. If this occurs, review your data for problem characters and replace them.

When Adaptive Server encounters a conversion error while sending data to the client, it replaces the bytes of the suspect characters with ASCII question marks (?). However, the query batch continues to completion. When the statement is complete, Adaptive Server sends the following message:

```
Msg 2403, Severity 16 (EX_INFO):
WARNING! Some character(s) could not be converted
into client's character set. Unconverted bytes were
changed to question marks ('?').
```

Conversions and changes to data lengths

In some cases, converting data between the server's character set and the client's character set results in a change to the length of the data. For example, this occurs when the character set on one system uses one byte to represent each character and the character set on the other system requires two bytes per character.

When character set conversion results in a change in data length, there are two possibilities:

- The data length decreases, as in the following examples:
  - Greek or Russian in multibyte UTF-8 to a single-byte Greek or Russian character set
  - Japanese two-byte Hankaku Katakana characters in EUC-JIS to single-byte characters in Shift-JIS
- The data length increases, as in the following examples:
  - Single-byte Thai to multibyte Thai in UTF-8
  - Single-byte Japanese characters in Shift-JIS to two-byte Hankaku Katakana in EUC-JIS

Configuring your system and application

If you are using UTF-8 anywhere in your client/server system, or using a Japanese character set, you are likely to encounter changes in data length as a result of character set conversion. If either of these conditions is true, you must configure your server to handle changes in data length. You may also need to set up your client to handle changes in data length.

1 Configure the server to use Unicode conversions. See "Configuring the server" on page 299. If the data length increases between the server and the client, then you must also complete steps 2 and 3.

2 The client must be using Open Client 11.1 or later. It must inform the server that it is able to handle CS_LONGCHAR data at connection time, using the Open Client ct_capability function.

   The *capability* parameter must be set to CS_DATA_LCHAR and the *value* parameter must be set to CS_TRUE:

   ```
   CS_INT capval = CS_TRUE
   ct_capability(connection,CS_SET,CS_CAP_RESPONS,
        CS_DATA_LCHAR,&capval)
   ```

   where *connection* is a pointer to a CS_CONNECTION structure.

3 When conversions result in an increase in data length, char and varchar data are converted to the client's character set and are sent to the client as CS_LONGCHAR data. The client application must be coded to extract the data received as CS_LONGCHAR.

Specifying the character set for utility programs

The Sybase utility programs assume that the default character set of the client platform is the same character set the client is using. However, sometimes the client character set differs from the character set for the platform. For this reason, you may need to specify the client character set at the command line. Character set conversion can be controlled in the standalone utilities. A command line option for the isql, bcp, and defncopy utilities specifies the client's character set and temporarily overrides settings of the LANG variable or settings in *locales.dat*.

-J *charset_name* (UNIX and PC) sets the client's character set to the *charset_name*.

Omitting the client character set's command-line flag causes the platform's default character set to be used. See the *Utility Guide* for information.

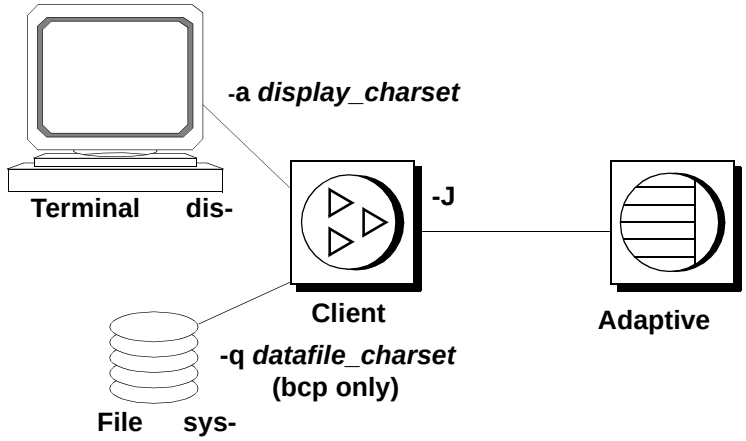
Display and file character set command-line options

Although the focus of this chapter is on character set conversion between client and Adaptive Server, there are two other places where you may need character set conversion:

- Between the client and a terminal
- Between the client and a file system

Figure 8-4 illustrates the paths and command-line options that are available in the standalone utilities isql, bcp, and defncopy.

Figure 8-4: Where character set conversion may be needed



As described earlier, the -J or /clientcharset command-line option specifies the character set used by the client when it sends and receives character data to and from Adaptive Server.

Setting the display character set

Use the -a command-line option if you are running the client from a terminal with a character set that differs from the client character set. In Figure 8-4, the -a option and the -J option are used together to identify the character set translation file (*.xlt* file) needed for the conversion.

Use -a without -J only if the client character set is the same as the default character set.

Setting the file character set

Use the -q command-line option if you are running bcp to copy character data to or from a file system that uses a character set that differs from the client character set. In Figure 8-4, use the -q or /filecharset option and the -J or /clientcharset option together to identify the character set translation file (*.xlt* file) needed for the conversion.

CHAPTER 9

Security Administration

This chapter provides an overview of the security features available in Adaptive Server.

Topics covered in this chapter include:

Security features available in Adaptive Server

SQL Server release 11.0.6 passed the security evaluation by the National Security Agency (NSA) at the Class C2 criteria. (The requirements for the C2 criteria are given by the Department of Defense in DOD 52.00.28-STD, *Department of Defense Trusted Computer System Evaluation Criteria* [TCSEC], also known as the “Orange Book.”)

The configuration of SQL Server release 11.0.6 that was evaluated at the C2 security level by the NSA in 1996 on the HP 9000 HP-UX BLS, 9.09+ platform is referred to as the evaluated configuration. Certain features of SQL Server, such as remote procedures and direct updates to system tables, were excluded from the evaluated configuration. Notes in the Adaptive Server documentation indicate particular features that were not included in the evaluated configuration. For a complete list of features that were excluded from the evaluated configuration, see Appendix A in the *SQL Server Installation and Configuration Guide for HP 9000 HP-UX BLS, 9.09+*.

Adaptive Server release 11.5 contains all of the security features included in SQL Server release 11.0.6 plus some new security features. Table 9-1 summarizes the major features.

Table 9-1: Major Security Features

Security feature	Description
Discretionary Access Controls (DAC)	Provides access controls that give object owners the ability to restrict access to objects, usually with the grant and revoke commands. This type of control is dependent upon an object owner's discretion.
Identification and authentication controls	Ensures that only authorized users can log into the system.
Division of roles	Allows you to grant privileged roles to specified users so that only designated users can perform certain tasks. Adaptive Server has predefined roles, called "system roles," such as System Administrator and System Security Officer. In addition, Adaptive Server allows System Security Officers to define additional roles, called "user-defined roles."
Network-based security	Provides security services to authenticate users and protect data transmitted among machines on a network.
Auditing	Provides the capability to audit events such as logins, logouts, server boot operations, remote procedure calls, accesses to database objects, and all actions performed by a specific user or with a particular role active. In addition, Adaptive Server provides a single option to audit a set of server-wide security-relevant events.

General process of security administration

Table 9-2 describes the major tasks that are required to administer Adaptive Server in a secure manner and refers you to the documentation that contains the instructions for performing each task.

Table 9-2: General process for security administration

Task	Description	See
1. Install Adaptive Server, including auditing.	This task includes preparing for installation, loading files from your distribution medium, performing the actual installation, and administering the physical resources that are required.	The the installation documentation for your platform
2. Set up a secure administrative environment.	This includes enabling auditing, granting roles to individual users to ensure individual accountability, and assigning login names to System Administrators and System Security Officers.	Chapter 6, “Managing Adaptive Server Logins and Database Users”
3. Add user logins to the server; add users to databases; establish groups and roles; set proxy aurhorizations.	Add logins, create groups, add users to databases, drop and lock logins, and assign initial passwords. Assign roles to users, create user-defined roles, and define role hierarchies and mutual exclusivity of roles.	Chapter 6, “Managing Adaptive Server Logins and Database Users”
4. Administer permissions for users, groups, and roles.	Grant and revoke permissions for certain SQL commands, executing certain system procedures, and accessing databases, tables, particular table columns, and views.	Chapter 7, “Managing User Permissions”
5. Administer the use of remote servers.	Establish and administer the access that is permitted between servers, add and drop remote server access, and map remote login names to local login names.	Chapter 9, “Managing Remote Servers” and the Adaptive Server installation and configuration documentation for your platform
6. Set up and maintain auditing.	Determine what is to be audited, audit the use of Adaptive Server, and use the audit trail to detect penetration of the system and misuse of resources.	Chapter 8, “Auditing” and the Adaptive Server installation and configuration documentation for your platform
7. Set up your installation for network-based security services.	Configure the server to use services, such as unified login, data confidentiality with encryption, data integrity, and determine security for remote procedures.	Chapter 10, “Using Network-Based Security”

Guidelines for setting up security

Use the guidelines described in the following sections when you set up security on Adaptive Server.

Using the “sa” login

When Adaptive Server is installed, a single login called “sa” is configured with the System Administrator and System Security Officer roles. This means that the “sa” login has unlimited power.

Use the “sa” login only during initial setup. Instead of allowing several users to use the “sa” account, establish individual accountability by assigning specific roles to individual administrators.

Warning! When logging in to Adaptive Server, do not use the -P option of isql to specify your password because another user may have an opportunity to see it.

Changing the “sa” Login Password

The “sa” login is configured initially with a “NULL” password. Use sp_password to change the password immediately after installation.

When to enable auditing

Enable auditing early in the administration process so that you have a record of privileged commands that are executed by System Security Officers and System Administrators. You might also want to audit commands that are executed by those with other special roles, such as operators when they dump and load databases.

Assigning login names

Assign Adaptive Server login names that are the same as their respective operating system login names. This makes logging in to Adaptive Server easier, simplifies management of server and operating system login accounts, and makes it easier to correlate the audit data generated by Adaptive Server with that of the operating system.

An Example of setting up security

Suppose you have decided to assign special roles to the users listed in Table 9-3.

Table 9-3: Users to whom you will assign roles

Name	Role	Operating system login name
Rajnish Smith	sso_role	rsmith
Catharine Macar-Swan	sa_role	cmacar
Soshi Ikedo	sa_role	sikedo
Julio Rozanski	oper_role	jrozan

Table 9-4 shows the sequence of commands you might use to set up a secure operating environment for Adaptive Server, based upon the role assignments shown in Table 5-3. After logging in to the operating system, you would issue these commands using the initial "sa" account.

Table 9-4: Examples of commands used to set up security

Commands	Result
isql -Usa	Logs in to Adaptive Server as "sa". Both sa_role and sso_role are active.
sp_audit "security", "all", "all", "on" sp_audit "all", "sa_role", "all", "on" sp_audit "all", "sso_role", "all", "on"'	Sets auditing options for server-wide, security-relevant events and the auditing of all actions that have sa_role or sso_role active.
sp_configure "auditing", 1	Enables auditing. Note: Before you enable auditing, set up a threshold procedure for the audit trail and determine how to handle the transaction log in sybsecurity. For details, see Chapter 12, "Auditing."
sp_addlogin rsmith, js&2P3d, @fullname = "Rajnish Smith" sp_addlogin cmacar, Fr3ds#1, @fullname = "Catharine Macar-Swan" sp_addlogin sikedo, mi5pd1s, @fullname = "Soshi Ikedo" sp_addlogin jrozan, w1seCrkr, @fullname = "Julio Rozanski"	Adds logins and passwords for Rajnish, Catharine, Soshi, and Julio. A default database is not specified for any of these users, so their default database is master.

Commands	Result
grant role sso_role to rsmith grant role sa_role to sikedo grant role sa_role to cmacar grant role oper_role to jrozan	Grants the sso_role to Rajnish, the sa_role to Soshi and Catharine, and the oper_role to Julio.
use sybsecurity sp_changedbowner rsmith	Grants access to the auditing database, sybsecurity, by making Rajnish, who is the System Security Officer, the database owner.
sp_locklogin sa,"lock"	Locks the “sa” login so that no one can log in as “sa”. Individuals can assume only the roles that are configured for them. Note: Do not lock the “sa” login until you have granted individual users the sa_role and sso_role roles and have verified that the roles operate successfully.

Discretionary access controls

Owners of objects can grant access to those objects to other users. Object owners can also grant other users the ability to pass the access permission to other users. With Adaptive Server’s discretionary access controls, you can give various kinds of permissions to users, groups, and roles with the grant command. Use the revoke command to rescind these permissions. The grant and revoke commands give users permission to execute specified commands and to access specified tables, views, and columns.

Some commands can be used at any time by any user, with no permission required. Others can be used only by users of a certain status such as a System Administrator and are not transferable.

The ability to assign permissions for the commands that can be granted and revoked is determined by each user’s status (as System Administrator, Database Owner, or database object owner), and by whether or not a particular user has been granted a permission with the option to grant that permission to other users.

Discretionary access controls are discussed in Chapter 7, “Managing User Permissions.”

Identification and authentication controls

Each Adaptive Server user is given a login account with a unique ID. All of that user's activity on the server can be attributed to a server user ID and audited.

Adaptive Server passwords are stored in the master..syslogins table in encrypted form. When you log into Adaptive Server from a client, you can choose client-side password encryption to encrypt your password before sending it over the network.

A System Security Officer can grant a user the ability to impersonate another user in the server. This ability, called **proxy authorization**, allows administrators to check permissions for a particular user or to perform maintenance on a user's database objects. Application servers can log in to the server and execute procedures and commands on behalf of several users.

Identification and authentication controls with network based security

Adaptive Server allows users to be pre-authenticated by a security mechanism before they log in to the server. This capability, called **unified login**, enables a user to log in to several servers without having to supply a login name and password for every connection.

Identification and authentication controls are discussed in Chapter 10, "Managing Adaptive Server Logins and Database Users." In addition, see "Using proxy authorization" and Chapter 13, "Managing Remote Servers."

Division of roles

An important feature in Adaptive Server is the division of *roles*. The roles supported by Adaptive Server enable you to enforce and maintain individual accountability. Adaptive Server provides system roles, such as System Administrator and System Security Officer, and user-defined roles, which are created by a System Security Officer.

Roles provide individual accountability for users performing operational and administrative tasks. Their actions can be audited and attributed to them.

Role hierarchy

A System Security Officer can define role hierarchies such that if a user has one role, the user automatically has roles lower in the hierarchy. For example, the “chief_financial_officer” role might contain both the “financial_analyst” and the “salary_administrator” roles. The Chief Financial Analyst can perform all tasks and see all data that can be viewed by the Salary Administrators and Financial Analysts.

Mutual exclusivity

Two roles can be defined to be mutually exclusive for:

- Membership – a single user cannot be granted both roles. For example, an installation might not want a single user to have both the “payment_requestor” and “payment_approver” roles to be granted to the same user.
- Activation – a single user cannot activate, or enable, both roles. For example, a user might be granted both the “senior_auditor” and the “equipment_buyer” roles, but the installation may not want to permit the user to have both roles enabled at the same time.

System roles, as well as user-defined roles, can be defined to be in a role hierarchy or to be mutually exclusive. For example, you might want a “super_user” role to contain the System Administrator, Operator, and “Tech Support” roles. In addition, you might want to define the System Administrator and System Security Officer roles to be mutually exclusive for membership; that is, a single user cannot be granted both roles.

See “Creating and assigning roles to users” on page 355 for information on administering and using roles.

Secure Sockets Layer (SSL) in Adaptive Server

Adaptive Server Enterprise security services now support secure sockets layer (SSL) session-based security. **SSL** is the standard for securing the transmission of sensitive information, such as credit card numbers, stock trades, and banking transactions, over the Internet.

While a comprehensive discussion of public-key cryptography is beyond the scope of this document, the basics are worth describing so that you have an understanding of how SSL secures Internet communication channels. This document is not a comprehensive guide to public-key cryptography.

The implementation of Adaptive Server SSL features assume that there is a knowledgeable System Security Officer who is familiar with the security policies and needs of your site, and who has general understanding of SSL and public-key cryptography.

Internet communications overview

TCP/IP is the primary transport protocol used in client/server computing, and is the protocol that governs the transmission of data over the Internet. TCP/IP uses intermediate computers to transport data from sender to recipient. The intermediate computers introduce weak links to the communication system where data may be subjected to tampering, theft, eavesdropping, and impersonation.

Public-key cryptography

Several mechanisms, known collectively as **public-key cryptography**, have been developed and implemented to protect sensitive data during transmission over the Internet. Public-key cryptography consists of encryption, key exchange, digital signatures, and digital certificates.

Encryption

Encryption is a process wherein a cryptographic algorithm is used to encode information to safeguard it from anyone except the intended recipient. There are two types of keys used for encryption:

- **Symmetric-key encryption** is where the same algorithm (key) is used to encrypt and decrypt the message. This form of encryption provides minimal security because the key is simple, and therefore easy to decipher. However, transfer of data that is encrypted with a symmetric key is fast because the computation required to encrypt and decrypt the message is minimal.

- **Public/private key encryption**, also known as asymmetric-key, are a pair of keys that are made up of public and private components to encrypt and decrypt messages. Typically, the message is encrypted by the sender with a private key, and decrypted by the recipient with the sender's public key, although this may vary. You can use a recipient's public key to encrypt a message, who then uses his private key to decrypt the message.

  The algorithms used to create public and private keys are more complex, and therefore harder to decipher. However, public/private key encryption requires more computation, sends more data over the connection, and noticeably slows data transfer.

Key exchange

The solution for reducing computation overhead and speeding transactions without sacrificing security is to use a combination of both symmetric key and public/private key encryption in what is known as a key exchange.

For large amounts of data, a symmetric key is used to encrypt the original message. The sender then uses either his private key or the recipient's public key to encrypt the symmetric key. Both the encrypted message and the encrypted symmetric key are sent to the recipient. Depending on what key was used to encrypt the message (public or private) the recipient uses the opposite to decrypt the symmetric key. Once the key has been exchanged, the recipient uses the symmetric key to decrypt the message.

Digital signatures

Digital signatures are used for tamper detection and non-repudiation. Digital signatures are created with a mathematical algorithm that generates a unique, fixed-length string of numbers from a text message; the result is called a hash or message digest.

To ensure message integrity, the message digest is encrypted by the signer's private key, then sent to the recipient along with information about the hashing algorithm. The recipient decrypts the message with the signer's public key. This process also regenerates the original message digest. If the digests match, the message proves to be intact and tamper free. If they do not match, the data has either been modified in transit, or the data was signed by an imposter.

Further, the digital signature provides **non-repudiation**—senders cannot deny, or repudiate, that they sent a message, because their private key encrypted the message. Obviously, if the private key has been compromised (stolen or deciphered), the digital signature is worthless for non-repudiation.

Certificates

Certificates are like passports: once you have been assigned one, the authorities have all your identification information in the system. Immigration control can access your information as you travel from country to country. Like a passport, the certificate is used to verify the identity of one entity (server, router, Web sites, and so on) to another.

Adaptive Server uses two types of certificates:

- **Server certificates** – a server certificate authenticates the server that holds it. Certificates are issued by a trusted third-party Certificate Authority (CA). The CA validates the holder's identity, and embeds the holder's public key and other identification information into the digital certificate. Certificates also contain the digital signature of the issuing CA, verifying the integrity of the data contained therein and validating its use.
- **CA certificates** (also known as **trusted root certificates) –** is a list of trusted CAs loaded by the server at start-up. CA certificates are used by servers when they function as a client, such as during remote procedure calls (RPCs). Adaptive Server loads its CA trusted root certificate at start-up. When connecting to a remote server for RPCs, Adaptive Server verifies that the CA that signed the remote server's certificate is a "trusted" CA listed in its own CA trusted roots file. If it is not, the connection fails.

Certificates are valid for a period of time and can be revoked by the CA for various reasons, such as when a security breach has occurred. If a certificate is revoked during a session, the session connection continues. Subsequent attempts to log in fail. Likewise, when a certificate expires, login attempts fail.

The combination of these mechanisms protect data transmitted over the Internet from eavesdropping and tampering. These mechanisms also protect users from impersonation, where one entity pretends to be another (spoofing), or where a person or an organization says it is set up for a specific purpose when the real intent is to capture private information (misrepresentation).

SSL overview

SSL is an industry standard for sending wire- or socket-level encrypted data over secure network connections.

Before the SSL connection is established, the server and the client exchange a series of I/O round trips to negotiate and agree upon a secure encrypted session. This is called the SSL handshake.

SSL handshake

When a client requests a connection, the SSL-enabled server presents its certificate to prove its identity before data is transmitted. Essentially, the handshake consists of the following steps:

- The client sends a connection request to the server. The request includes the SSL (or Transport Layer Security, TLS) options that the client supports.
- The server returns its certificate and a list of supported CipherSuites, which includes SSL/TLS support options, algorithms used for key exchange, and digital signatures.
- A secure, encrypted session is established when both client and server have agreed upon a CipherSuite.

For more specific information about the **SSL handshake** and the SSL/TLS protocol, see the Internet Engineering Task Force Web site at http://www.ietf.org.

For a list of CipherSuites that Adaptive Server supports, see "CipherSuites" on page 328.

SSL in Adaptive Server

Adaptive Server's implementation of SSL provides several levels of security.

- The server authenticates itself—proves that it is the server you intended to contact—and an encrypted SSL session begins before any data is transmitted.
- Once the SSL session is established, the client requesting a connection can send his user name and password over the secure, encrypted connection.

- A comparison of the digital signature on the server certificate can determine whether the data received by the client was modified before reaching the intended recipient.

Adaptive Server uses the SSL Plus™ library API from Certicom Corp.

SSL filter

Adaptive Server's directory service, such as the interfaces file, NT registry, or LDAP service, defines the server address and port numbers, and determines the security protocols that are enforced for client connections. Adaptive Server implements the SSL protocol as a filter that is appended to the master and query lines of the directory services.

The addresses and port numbers on which Adaptive Server accepts connections are configurable so that multiple network and security protocols can be enabled for a single server. Server connection attributes are specified with directory services, such as LDAP or DCE, or with the traditional Sybase interfaces file. See "Creating server directory entries" on page 325.

All connection attempts to a master or query entry in the interfaces file with an **SSL filter** must support the SSL protocol. A server can be configured to accept SSL connections and have other connections that accept clear text (unencrypted data), or use other security mecahnisms.

For example, the interfaces file on UNIX that supports both SSL-based connections and clear-text connections looks like:

```
SYBSRV1
    master tli tcp /dev/tcp \x00020abc12345678000000000000000 ssl
    query tli tcp /dev/tcp \x00020abc12345678000000000000000 ssl
    master tli tcp /dev/tcp \x00020abd12345678000000000000000
```

The SSL filter is different from other security mechanisms, such as DCE and Kerberos, which are defined with SECMECH (security mechanism) lines in the interfaces file (*sql.ini* on Windows).

Authentication via the certificate

The SSL protocol requires server authentication via a server certificate to enable an encrypted session. Likewise, when Adaptive Server is functioning as a client during RPCs there must be a repository of trusted CAs that a client connection can access to validate the server certificate.

The server certificate

Each Adaptive Server must have its own server certificate file that is loaded at start-up. The default location for the certificates file is:

UNIX – *$SYBASE/$SYBASE_ASE/certificates/servername.crt*

NT – *%SYBASE%\%SYBASE_ASE%\certificates\servername.crt*

where *servername* is the name of the Adaptive Server as specified on the command line during start-up with the -s flag, or from the environment variable $DSLISTEN.

The server certificate file consists of encoded data, including the server's certificate and the encrypted private key for the server certificate.

Alternatively, you can specify the location of the server certificate file when using sp_ssladmin.

Note To make a successful client connection, the common name in the certificate must match the Adaptive Server name in the interfaces file.

The CA trusted roots certificate

The list of trusted CAs is loaded by Adaptive Server at start-up from the trusted roots file. The trusted roots file is similar in format to a certificate file, except that it contains certificates for CAs known to Adaptive Server. A trusted roots file is accessible by the local Adaptive Server in:

UNIX – *$SYBASE/$SYBASE_ASE/certificates/servername.txt*

NT – *%SYBASE%\%SYBASE_ASE\certificates\servername.txt*

where *servername* is the name of the server. The trusted roots file is only used by Adaptive Server when it is functioning as a client, such as when performing (RPC) calls or Component Integration Services (CIS) connections.

The System Security Officer adds and deletes CAs that are to be accepted by Adaptive Server, using a standard ASCII-text editor.

Warning! Use the System Security Officer role (sso_role) within Adaptive Server to restrict access and execution on security-sensitive objects.

Adaptive Server provides tools to generate a certificate request and to authorize certificates. See "Using Adaptive Server tools to request and authorize certificates" on page 324.

Connection types

This section describes various client-to-server and server-to-server connections.

Client login to Adaptive Server

Open Client applications establish a socket connection to Adaptive Server similarly to the way that existing client connections are established. Before any user data is transmitted, an SSL handshake occurs on the socket when the network transport-level connect call completes on the client side and the accept call completes on the server side.

Server-to-server remote procedure calls

Adaptive Server establishes a socket connection to another server for RPCs in the same way that existing RPC connections are established. Before any user data is transmitted, an SSL handshake occurs on the socket when the network transport-level connect call completes. If the server-to-server socket connection has already been established, then the existing socket connection and security context is reused.

When functioning as a client during RPCs, Adaptive Server requests the remote server's certificate during connection. Adaptive Server then verifies that the CA that signed the remote server's certificate is trusted; that is to say, on its own list of trusted CAs in the trusted roots file. It also verifies that the common name in the server certificate matches the common name used when establishing the connection.

Companion Server and SSL

You can use a companion server to configure Adaptive Server for failover. You must configure both the primary and secondary servers with the same SSL and RPC configuration. When connections fail over or fail back, security sessions are reestablished with the connections.

Open Client connections

Component Integration Services, RepAgent, Distributed Transaction Management, and other modules in Adaptive Server use Client-Library to establish connections to servers other than Adaptive Server. The remote server is authenticated by its certificate. The remote server authenticates the Adaptive Server client connection for RPCs with user name and password.

Enabling SSL

Adaptive Server determines which security service it will use for a port based on the interface file (*sql.ini* on Windows).

To enable SSL:

1 Generate a certificate for the server.

2 Create a trusted roots file.

3 Use sp_configure to enable SSL. From a command prompt, enter:

```
sp_configure "enable ssl", 1
```

1 enables the SSL subsystem at start-up, allocates memory, and SSL performs wire-level encryption of data across the network.

0 disables SSL. This value is the default.

4 Add the SSL filter to the interfaces file. See "Creating server directory entries" on page 325.

5 Use sp_ssladmin to add a certificate to the certificates file. See "Administering certificates" on page 326.

6 Shut down and restart Adaptive Server.

Note To request, authorize, and convert third-party certificates, see the *Adaptive Server Utilities Guide* for information on certauth, certreq, and certpk12 tools.

Unlike other security services, such as DCE, Kerberos, and NTLAN, SSL relies neither on the "Security" section of the Open Client/Open Server configuration file *libtcl.cfg*, nor objects in *objectid.dat*.

The System Administrator should consider memory use by SSL when planning for total physical memory. You will need approximately 40K per connection (connections include user connections, remote servers, and network listeners) in Adaptive Server for SSL connections. The memory is reserved and preallocated within a memory pool and is used internally by Adaptive Server and SSL Plus libraries as requested.

Obtaining a certificate

The System Security Officer installs server certificates and private keys for Adaptive Server by:

- Using third-party tools provided with existing public-key infrastructure already deployed in the customer environment.
- Using the Adaptive Server certificate request tool in conjunction with a trusted third-party CA.

To obtain a certificate, you must request a certificate from a CA. If you request a certificate from a third-party and that certificate is in PKCS #12 format, use the certpk12 utility to convert the certificate into a format that is understood by Adaptive Server.

To test the Adaptive Server certificate request tool and to verify that the authentication methods are working on your server, Adaptive Server provides a tool, for testing purposes, that allows you to function as a CA and issue CA-signed certificate to yourself.

The main steps to creating a certificate for use with Adaptive Server are:

1 Generate the public and private key pair.

2 Securely store the private key.

3 Generate the certificate request.

4 Send the certificate request to the CA.

5 After the CA signs and returns the certificate, store it in a file and append the private key to the certificate.

6 Store the certificate in the Adaptive Server installation directory.

Third-party tools to request certificates

Most third-party PKI vendors and some browsers have utilities to generate certificates and private keys. These utilities are typically graphical wizards that prompt you through a series of questions to define a distinguished name and a common name for the certificate.

Follow the instructions provided by the wizard to create certificate requests. Once you receive the signed PKCS #12-format certificate, use certpk12 to generate a certificate file and a private key file. Concatenate the two files into a *servername.crt* file, where *servername* is the name of the server, and place it in the *certificates* directory under *$SYBASE/$SYBASE_ASE*. See the *Adaptive Server Utilities Guide* for your platform.

Using Adaptive Server tools to request and authorize certificates

Adaptive Server provides two tools for requesting and authorizing certificates. certreq generates public and private key pairs and certificate requests. certauth converts a server certificate request to a CA-signed certificate.

Warning! Use certauth only for testing purposes. Sybase recommends that you use the services of a commercial CA because it provides protection for the integrity of the root certificate, and because a certificate that is signed by a widely accepted CA facilitates the migration to the use of client certificates for authentication.

Preparing the server's trusted root certificate is a five-step process. Perform the first two steps to create a test trusted root certificate so you can verify that you are able to create server certificates. Once you have a test CA certificate (trusted roots certificate) repeat steps three through five to sign server certificates.

1. Use certreq to request a certificate.
2. Use certauth to convert the certificate request to a CA self-signed certificate (trusted root certificate).
3. Use certreq to request a server certificate and private key.
4. Use certauth to convert the certificate request to a CA-signed server certificate.
5. Append the private key text to the server certificate and store the certificate in the server's installation directory.

For information about Sybase utilities, certauth, certreq, and certpk12 for requesting, authorizing and converting third-party certificates, see the *Adaptive Server Utilities Guide* for your platform.

Note certauth and certreq are dependent on RSA and DSA algorithms. These tools only work with crypto modules that use RSA and DSA algorithms to construct the certificate request.

Adaptive Server supports the Certicom Corp. cryptographic engine, Security Builder™, which supports RSA and DSA algorithms to construct the certificate requests.

Creating server directory entries

Adaptive Server accepts client logins and server-to-server RPCs. The address and port numbers where Adaptive Server accepts connections are configurable so you can specify multiple networks, different protocols, and alternate ports.

In the interfaces file, SSL is specified as a filter on the master and query lines, whereas security mechanisms such as DCE or Kerberos are identified with a SECMECH line. The following example shows a TLI-based entry for an Adaptive Server using SSL in a UNIX environment:

An entry for an Adaptive Server with SSL and DCE security mechanisms on UNIX might look like:

```
SYBSRV1
    master tli tcp /dev/tcp \x00020abc12345678000000000000000 ssl
    query tli tcp /dev/tcp \x00020abc12345678000000000000000 ssl
    master tli tcp /dev/tcp \x00020abd12345678000000000000000
    SECMECH 1.3.6.1.4.897.4.6.1
```

An entry for the server with SSL and Kerberos security mechanisms on NT might look like:

```
[SYBSRV2]
    query=nlwnsck, 18.52.86.120,2748,ssl
    master=nlwnsck 18.52.86.120,2748,ssl
    master=nlwnsck 18.52.86.120,2749
    secmech=1.3.6.1.4.897.4.6.6
```

The SECMECH lines for SYBSRV1 and SYBSRV2 in the examples contain an object identifier (OID) that refers to security mechanisms DCE and Kerberos, respectively. The OID values are defined in:

UNIX – *$SYBASE/$SYBASE_OCS/config/objectid.dat*

NT – *%SYBASE%\%SYBASE_OCS\ini\objectid.dat*

In these examples, the SSL security service is specified on port number 2748(0x0abc).

Note The use of SSL concurrently with a SECMECH security mechanism is intended to facilitate migration from SECMECHs to SSL security.

Administering certificates

To administer SSL and certificates in Adaptive Server, use sp_ssladmin. sso_role is required to execute the stored procedure.

The sp_ssladmin is used to:

- Add local server certificates. You can add certificates and specify the password used to encrypt private keys, or require input of the password at the command line during start-up.
- Delete local server certificates.
- List server certificates.

The syntax for sp_ssladmin is:

```
sp_ssladmin {[addcert, certificate_path [, password|NULL]]
      [dropcert, certificate_path]
      [lscert]
      [help]}
```

For example:

```
sp_ssladmin addcert, "/sybase/ASE-12_5/certificates/Server1.crt",
      "mypassword"
```

This adds an entry for the local server, *Server1.crt*, in the certificates file in the absolute path to */sybase/ASE-12_5/certificates* (*x:\sybase\ASE-12_5\certificates* on Windows). The private key is encrypted with the password "*mypassword*". The password should be the one specified when you created the private key.

Before accepting the certificate, sp_ssladmin verifies that:

- The private key can be decrypted using the provided password (except when NULL is specified).
- The private key and public key in the certificate match.
- The certificate chain, from root CA to the server certificate, is valid.
- The common name in the certificate matches the common name in the interfaces file.

If the common names do not match, sp_ssladmin issues a warning. If the other criteria fails, the certificate is not added to the certificates file.

Warning! Adaptive Server limits passwords to 64 characters. In addition, certain platforms restrict the length of valid passwords when creating server certificates. Select a password within these limits:

- Sun Solaris – both 32- and 64-bit platforms, maximum 256 characters.
- Linux – 128 characters.
- IBM – both 32- and 64-bit platforms, 32 characters.
- HP – both 32- and 64-bit platforms, 8 characters.
- Digital UNIX – 80 characters.
- Windows NT – 256 characters.

The use of NULL as the password is intended to protect passwords during the initial configuration of SSL, before the SSL encrypted session begins. Since you have not yet configured SSL, the password travels unecrypted over the connection. You can avoid this by specifying the password as NULL during the first log in.

When NULL is the password, you must start dataserver with a -y flag, which prompts the administrator for the private-key password at the command line.

After restarting Adaptive Server with an SSL connection established, use sp_ssladmin again, this time using the actual password. The password is then encrypted and stored by Adaptive Server. Any subsequent starts of Adaptive Server from the command line use the encrypted password; you do not have to specify the password on the command line during start-up.

An alternative to using a NULL password during the first login, is to avoid a remote connection to Adaptive Server via isql. You can specify “localhost” as the *hostname* in the *interfaces* file (*sql.ini* on Windows) to prevent clients from connecting remotely. Only a local connection can be established, and the password is never transmitted over a network connection.

Performance

There is additional overhead required to establish a secure session, because data increases in size when it is encrypted, and it requires additional computation to encrypt or decrypt information. Typically, the additional I/O accrued during the SSL handshake may make user login 10 to 20 times slower. Also, SSL-enabled connections require more memory. You must have approximately 40K more memory for each user connection.

CipherSuites

During the SSL handshake, the client and server negotiate a common security protocol via a CipherSuite. **CipherSuites** are preferential lists of key-exchange algorithms, hashing methods, and encryption methods used by SSL-enabled applications. For a complete description of CipherSuites, visit the Internet Engineering Task Force (IETF) organization at http://www.ietf.org/rfc/rfc2246.txt.

By default, the strongest CipherSuite supported by both the client and the server is the CipherSuite that is used for the SSL-based session.

Adaptive Server supports the CipherSuites that are available with the SSL Plus library API and the cryptographic engine, Security Builder™, both from Certicom Corp.

The following lists the CipherSuites, ordered by strength from stronget to weakest, supported in Adaptive Server 12.5.

```
TLS_RSA_WITH_3DES_EDE_CBC_SHA,
TLS_RSA_WITH_RC4_128_SHA,
TLS_RSA_WITH_RC4_128_MD5,
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA,
TLS_DHE_DSS_WITH_RC4_128_SHA,
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA,
TLS_RSA_WITH_DES_CBC_SHA,
TLS_DHE_DSS_WITH_DES_CBC_SHA,
TLS_DHE_RSA_WITH_DES_CBC_SHA,
TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA,
TLS_RSA_EXPORT1024_WITH_RC4_56_SHA,
TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA,
TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA,
TLS_RSA_EXPORT_WITH_RC4_40_MD5,
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA,
```

```
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
```

Note The CipherSuites listed above conform to the transport layer specification (TLS). TLS is an enhanced version of SSL 3.0, and is an alias for the SSL version 3.0 CipherSuites.

Network-based security

Adaptive Server provides network-based security services that enable you to authenticate users and protect data transmitted among machines on a network.

In a distributed client/server computing environment intruders can view or tamper with confidential data. With Adaptive Server, you can use security services provided by third-party providers to authenticate users, encrypt data, and prevent data tampering.

Depending upon the security mechanism you choose, Adaptive Server allows you to use one or more of these security services:

- Unified login – use a security mechanism to authenticate users *once* without requiring them to supply a name and password every time they log in to an Adaptive Server.
- Message confidentiality – encrypt data over the network.
- Mutual authentication – use the security mechanism to verify the identity of the client and the server. (This must be requested by the client and cannot be required by Adaptive Server.)
- Message integrity – verify that data communications have not been modified.
- Replay detection – verify that data has not been intercepted by an intruder.
- Out-of-sequence check – verify the order of data communications.
- Message origin checks – verify the origin of the message.

- Remote procedure security – establish mutual authentication, message confidentiality, and message integrity for remote procedure communications.

Note The security mechanism you are using may not support all of these services.

Auditing

Adaptive Server includes a comprehensive audit system. The audit system consists of a system database called sybsecurity, configuration parameters for managing auditing, a system procedure, sp_audit, to set all auditing options, and a system procedure, sp_addauditrecord, to add user-defined records to the audit trail. When you install auditing, you can specify the number of audit tables that Adaptive Server will use for the audit trail. If you use two or more tables to store the audit trail, you can set up a smoothly running audit system with no manual intervention and no loss of records.

A System Security Officer manages the audit system and is the only user who can start and stop auditing, set up auditing options, and process the audit data. As a System Security Officer, you can establish auditing for events such as:

- Server-wide, security-relevant events
- Creating, deleting, and modifying database objects
- All actions by a particular user or all actions by users with a particular role active
- Granting or revoking database access
- Importing or exporting data
- Logins and logouts

Auditing functionality is discussed in Chapter 12, “Auditing.”

User-defined login security

UDLS gives you more control over security-related features of Adaptive Server.

In Adaptive Server versions 12.0 and later, the System Security Officer can:

- Add more user logins and roles than was possible in earlier versions
- Specify the maximum allowable number of times an invalid password can be entered for a login or role before that login or role is automatically locked
- Lock and unlock roles manually
- Ensure that all user passwords have at least one digit
- Specify the minimum password length required server-wide or for a specific login or role
- Display all security-related information for logins and roles
- Associate a password expiration value with a specified login or role

Negative values may be used for user IDs (*uid*).

The server user ID (s*uid*) associated with a group or a role in sysusers is not equal to the negation of their user ID (*uid*). Every *suid* associated with a group or a role in sysusers is set to -2 (INVALID_SUID).

Setting and changing the maximum login attempts

Setting the maximum number of login attempts allowed provides protection against "brute-force" or dictionary-based attempts to guess passwords. A System Security Officer can specify a maximum number of consecutive login attempts allowed, after which the login or role is automatically locked. The number of allowable failed login attempts can be set for the entire server or for individual logins and roles. Individual settings override the server-wide setting.

The number of failed logins is stored in the logincount column in master..syslogins. A successful login resets the number of failed logins to 0.

Setting the server-wide maximum allowed login attempts

To set the server-wide maximum number of login attempts for logins and roles, use the maximum failed logins configuration parameter.

For example:

```
sp_configure "maximum failed logins", 5
```

Sets the system-wide maximum number of failed login attempts to 5.

For details on the syntax and rules for using maximum failed logins, see sp_configure.

Setting the maximum allowed login attempts for specific logins

To set the maximum number of login attempts for a specific login at creation, use sp_addlogin.

For example:

```
sp_addlogin joe, "Djdiek3", pubs2, null, null, null, null, 2
```

Creates the new login joe with the password “Djdiek3” and sets the maximum number of failed login attempts for the login joe to 2.

For details on the syntax and rules for using maxfailedlogins, see sp_addlogin.

Setting the maximum allowed login attempts for specific roles

To set the maximum number of login attempts for a specific role at creation, use create role.

For example:

```
create role intern_role with passwd "temp244", max failed_logins 20
```

Creates the intern_role role with the password “temp244”, and sets the maximum number of failed login attempts for intern_role to 20.

For details on the syntax and rules for using max failed_logins, see create role.

Changing the maximum allowed login attempts for specific logins

Use sp_modifylogin to set or change the maximum failed login attempts for an existing login.

For example:

```
sp_modifylogin "joe", @option="max failed_logins", @value="40"
```

Changes the maximum number of failed login attempts for the login “joe” to 40.

Note The *value* parameter is a character datatype; therefore, quotes are required for numeric values.

```
sp_modifylogin "all overrides", "max failed_logins", "3"
```

Changes the overrides for maximum failed login attempts of all logins to 3.

```
sp_modifylogin "all overrides", @option="max failed_logins", @value="-1"
```

Removes the overrides for maximum failed logins option for all logins.

sp_modifylogin only effects user roles, not system roles. For details on the syntax and rules for using max failed_logins, see sp_modifylogin.

Changing the maximum allowed login attempts for specific roles

Use alter role to set or change the maximum failed login attempts for an existing role.

For example:

```
alter role physician_role set max failed_logins 5
```

Changes the maximum failed logins allowed for physician_role to 5.

```
alter role "all overrides" set max failed_logins -1
```

Removes the overrides for the maximum failed logins for all roles.

For details on the syntax and rules for using max failed_logins, see alter role.

Locking and unlocking logins and roles

A login or role can be locked when:

- Its password expires, or
- The maximum number of failed login attempts occur, or

- The System Security Officer locks the login or role manually.

Locking and unlocking logins

The System Security Officer can use sp_locklogin to lock or unlock a login manually. (This is not new functionality, but is mentioned here for comparison to the new methods available for locking and unlocking roles.)

For example:

```
sp_locklogin "joe" , "lock"
sp_locklogin "joe" , "unlock"
```

Information about the lock status of a login is stored in the status column of syslogins.

For details on the syntax and rules for using sp_locklogin, see sp_locklogin.

Locking and unlocking roles

The System Security Officer can use alter role to lock or unlock a role manually.

For example:

```
alter role physician_role lock
alter role physician_role unlock
```

Information about the lock status of a role is stored in the status column of syssrvroles.

For details on the syntax and rules for using lock and unlock, see alter role.

Unlocking logins and roles at server startup

Automatic login lockouts can cause a site to end up in a situation in which all accounts capable of unlocking logins (System Administrators and System Security Officers) are locked. In these situations, use the -u flag with the dataserver utility to unlock a specific login or role when you start Adaptive Server.

For details on the syntax and rules for using the -u flag, see the *Utility Guide*.

Displaying password information

This section discusses displaying password information for logins and roles.

Displaying password information for specific logins

Use sp_displaylogin to display the password settings for a login.

For example, the following statement displays information about the login joe:

```
sp_displaylogin joe
Suid: 2
Loginame: joe
Fullname: Joseph Resu
Default Database: master
Default Language:
Configured Authorization: intern_role (default OFF)
Locked: NO
Date of Last Password Change: Nov 24 1997  3:35PM
Password expiration interval : 5
Password expired : NO
Minimum password length:4
Maximum failed logins : 10
Current failed logins : 3
```

For details on the syntax and rules, see sp_displaylogin.

Displaying password information for specific roles

Use sp_displayroles to display the password settings for a role.

For example:

```
sp_displayroles physician_role, "display_info"
Role name = physician_role
Locked : NO
Date of Last Password Change : Nov 24 1997  3:35PM
Password expiration interval = 5
Password expired : NO
Minimum password length = 4
Maximum failed logins = 10
Current failed logins = 3
```

Displays information about the physician_role role.

For details on the syntax and rules, see sp_displayroles.

Checking passwords for at least one character

The System Security Officer can tell the server to check for at least one character or digit in a password using the server-wide configuration parameter, check password for digit. If set, this parameter does not affect existing passwords. By default, checking for digits is off.

For example:

```
sp_configure "check password for digit", 1
```

Activates the check password functionality.

```
sp_configure "check password for digit", 0
```

Deactivates the check password functionality.

For details on the syntax and rules for using the new parameter, see sp_configure.

Setting and changing minimum password length

In previous releases, the minimum password length was a non-configurable, hard-coded value of six characters. The configurable password allows you to customize passwords to fit your needs such as using four-digit personal identification numbers (PINs) or anonymous logins with NULL passwords.

The System Security Officer can specify:

- A globally enforced minimum password length
- A per-login or per-role minimum password length

The per-login or per-role value overrides the server-wide value. Setting a minimum password length affects only new passwords created after setting the value. It does not affect existing passwords.

Setting the server-wide minimum password length

Use the minimum password length configuration parameter to specify a server-wide value for minimum password length for both logins and roles.

For example:

```
sp_configure "minimum password length", 4
```

Sets the minimum password length for all logins and roles to four characters.

For details on the syntax and rules for using minimum password length, see sp_configure.

Setting minimum password length for a specific login

To set the minimum password length for a specific login at creation, use sp_addlogin.

For example:

```
sp_addlogin joe, "Djdiek3", @minpwdlen=4
```

Creates the new login joe with the password “Djdiek3”, and sets the minimum password length for joe to 4. d

For details on the syntax and rules for using minpwdlen, see sp_addlogin.

Setting minimum password length for a specific role

To set the minimum password length for a specific role at creation, use create role.

For example:

```
create role intern_role with passwd "temp244", min passwd length 0
```

Creates the new role intern_role with the password “temp244” and sets the minimum password length for intern_role to 0. The original password is seven characters, but the password can be changed to one of any length because the minimum password length is set to 0.

For details on the syntax and rules for using min passwd length, see create role.

Changing minimum password length for a specific login

Use sp_modifylogin to set or change the minimum password length for an existing login. sp_modifylogin only effects user roles, not system roles.

For example:

```
sp_modifylogin "joe", @option="min passwd length", @value="8"
```

Changes the minimum password length for the login “joe” to eight characters.

Note The *value* parameter is a character datatype; therefore, quotes are required for numeric values.

```
sp_modifylogin "all overrides", "min passwd length", @value="2"
```

Changes the value of the overrides for minimum password length of all logins to two characters.

```
sp_modifylogin "all overrides", @option="min passwd length", @value="-1"
```

Removes the overrides for the minimum password length for all logins.

For details on the syntax and rules for using min passwd length, see sp_modifylogin.

Changing minimum password length for a specific role

Use alter role to set or change the minimum password length for an existing role.

For example:

```
alter role physician_role set min passwd length 5
```

Sets the minimum length for physician_role, an existing role, to five characters.

```
alter role "all overrides" set min passwd length -1
```

Overrides the minimum password length of all roles.

For details on the syntax and rules for using set min passwd length, see alter role.

Setting the expiration interval for a password

System Administrators and System Security Officers can:

Use	To
sp_addlogin	Specify the expiration interval for a login password at creation

Use	To
sp_modifylogin	Change the expiration interval for a login password. sp_modifylogin only effects user roles, not system roles.
create role	Specify the expiration interval for a role password at creation
alter role	Change the expiration interval for a role password

The following rules apply to password expiration for logins and roles:

- A password expiration interval assigned to individual login accounts or roles overrides the global password expiration value. This allows you to specify shorter expiration intervals for sensitive accounts or roles, such as System Security Officer passwords, and more relaxed intervals for less sensitive accounts such as an anonymous login.
- A login or role for which the password has expired is not directly activated.

For details on the syntax and rules for the commands and system procedures, see the *Adaptive Server Reference Manual*.

Password expiration turned off for pre-12.x passwords

Password expiration did not affect roles in releases prior to Adaptive Server 12.x. Therefore, in Adaptive Server 12.x and later, password expiration is deactivated for any existing user-defined role passwords. During the upgrade all user-defined role passwords are stamped as having a password interval of 0.

Message for impending password expiration

When a password for a login or role is about to expire, a warning message asks the user to contact the System Security Officer.

Circumventing password protection

Circumventing the password-protection mechanism may be necessary in the case of automated login systems. You can create a role that could access other roles without passwords.

If a System Security Officer wants to bypass the password mechanism for certain users, the System Security Officer can grant the password-protected role to another role and grant this new role to one or more users. Activation of this role automatically activates the password-protected role without having to provide a password.

For example:

Jane is the System Security Officer for the fictitious company ABC Inc., which uses automated login systems. Jane creates the following roles:

- financial_assistant

  ```
  create role financial_assistant with passwd
  "L54K3j"
  ```

- accounts_officer

  ```
  create role accounts_officer with passwd "9sF6ae"
  ```

- chief_financial_officer

  ```
  create role chief_financial_officer
  ```

Jane grants the roles of financial_assistant and accounts_officer to the chief_financial_officer role:

```
grant role financial_assistant, accounts_officer to
chief_financial_officer
```

Jane then grants the chief_financial_officer role to Bob:

```
grant role chief_financial_officer to bob
```

Bob logs in to Adaptive Server and activates the chief_financial_officer role:

```
set role chief_financial_officer on
```

The roles of financial_assistant and accounts_officer are automatically activated without Bob providing a password. Bob now has the ability to access everything under the financial_assistant and accounts_officer roles without having to enter the passwords for those roles.

Creating a password expiration interval for a new login

Use sp_addlogin to set the password expiration interval for a new login.

For example:

```
sp_addlogin joe, "Djdiek3", null, null, null, 2
```

Creates the new login joe with the password “Djdiek3”, and sets the password expiration interval for joe to 2 days.

For details on the syntax and rules for using the new parameter, see sp_addlogin.

Creating a password expiration interval for a new role

Use create role to set the password expiration interval for a new role.

For example:

```
create role intern_role with passwd "temp244", passwd expiration 7
```

Creates the new role intern_role with the password “temp244”, and sets the password expiration interval for intern_role to 7 days.

For details on the syntax and rules for using passwd expiration, see create role.

Creation date added for passwords

Passwords are stamped with a “creation date” equal to the upgrade date of a given server. The creation date for login passwords is stored in the pwdate column of syslogins. The creation date for role passwords is stored in the pwdate column of syssrvroles.

Changing or removing password expiration interval for login or role

Use sp_modifylogin to change the password expiration interval for an existing login, add a password expiration interval to a login that did not have one, or remove a password expiration interval. sp_modifylogin only effects user roles, not system roles.

For example:

```
sp_modifylogin "joe", @option="passwd expiration", @value="5"
```

Changes the password expiration interval for the login “joe” to 5 days.

Note The *value* parameter is a character datatype; therefore, quotes are required for numeric values.

```
sp_modifylogin "all overrides", @option="passwd expiration", @value="3"
```

Changes the value of the overrides for the password expiration for all logins to 3 days.

```
sp_modifylogin "all overrides”, @option="passwd expiration", @value="-1"
```

Removes the value of the overrides for the password expiration for all logins.

For details on the syntax and rules for using passwd expiration, see sp_modifylogin.

CHAPTER 10

Managing Adaptive Server Logins and Database Users

This chapter describes methods for managing Adaptive Server login accounts and database users.

Topics covered in this chapter include:

Adding new users: An overview

The process of adding new logins to Adaptive Server, adding users to databases, and granting them **permission** to use commands and database objects is divided among the System Security Officer, System Administrator, and Database Owner.

Adding new users consists of the following steps:

1 A System Security Officer uses sp_addlogin to create a server login account for a new user.

2 A System Administrator or Database Owner uses sp_adduser to add a user to a database. This command can also give the user an alias or assign the user to a group. For more information, see "Creating groups" on page 347.

3 A System Security officer grants specific roles to the user.

4 A System Administrator, Database Owner, or object owner grants the user or group specific permissions on specific commands and database objects. Users or groups can also be granted permission to grant specific permissions on objects to other users or groups. See Chapter 11, "Managing User Permissions" for detailed information about permissions.

Table 10-1 summarizes the system procedures and commands used for these tasks.

Table 10-1: Adding users to Adaptive Server and databases

Task	Required role	Command or procedure	Database
Create new logins, assign passwords, default databases, default language, and full name	System Security Officer	sp_addlogin	Any database
Create groups	Database Owner or System Administrator	sp_addgroup	User database
Create and assign roles	System Security Officer	create role	
Add users to database, assign aliases, and assign groups	Database Owner or System Administrator	sp_adduser	User database
Grant groups, users, or roles permission to create or access database objects	Database Owner, System Administrator, or object owner	grant	User database

Choosing and creating a password

Your password helps prevent access by unauthorized people. When you create your password, follow these guidelines:

- Do not use information such as your birthday, street address, or any other word or number that has anything to do with your personal life.
- Do not use names of pets or loved ones.

- Do not use words that appear in the dictionary or words spelled backwards.

The most difficult passwords to guess are those that combine uppercase and lowercase letters and numbers. Never give anyone your password, and never write it down where anyone can see it.

Follow these rules to create a password:

- Passwords must be at least 6 bytes long.
- Passwords can consist of any printable letters, numbers, or symbols.
- A password must be enclosed in quotation marks in sp_addlogin if it:
  - Includes any character other than A-Z, a-z, 0-9,_, #, valid single-byte or multibyte alphabetic characters, or accented alphabetic characters
  - Begins with a number 0–9

Adding logins to Adaptive Server

Use sp_addlogin to add a new **login** name to Adaptive Server. You do not use it to give the user permission to access user databases. Use sp_adduser for that purpose. Only the System Security Officer can execute sp_addlogin. The syntax is:

```
sp_addlogin loginame, passwd [, defdb]
    [, deflanguage [, fullname]]]
```

where:

- *loginame* is the new user's login name. The login name must follow the rules for identifiers and must be unique on Adaptive Server. To simplify both the login process and server administration, make the Adaptive Server login name the same as the user's operating system login name. This makes logging in to Adaptive Server easier because many client programs use the operating system login name as a default. It also simplifies management of server and operating system login accounts, and makes it easier to correlate usage and audit data generated by Adaptive Server and by the operating system.

- *passwd* is the password for the new user. For guidelines on choosing and creating secure passwords, see "Choosing and creating a password" on page 344. For information on changing a password, see "Changing passwords" on page 369.
- *defdb* is the **default database** – where the user starts each session of Adaptive Server.

  Note The default database is master. To discourage users from creating database objects in the master database, assign a default database other than master to most users.

  A System Administrator can change anyone's default database with sp_modifylogin. Other users can change only their own default database.

  After specifying the default database, add the user to the default database with sp_adduser so that he or she can log in directly to the default database.
- *deflanguage* is the **default language** in which the user's prompts and messages are displayed. If you omit this parameter, Adaptive Server's default language is used. A System Administrator can change any user's default language with sp_modifylogin. Other users can change only their own language.
- *fullname* is the full name of the user. This is useful for documentation and identification purposes. If omitted, no full name is added. A System Administrator can change any user's full name with sp_modifylogin. Other users can change only their own full name.

The following statement sets up an account for the user "maryd" with the password "100cents," the default database (master), the default language, and no full name:

```
sp_addlogin "maryd", "100cents"
```

The password requires quotation marks because it begins with 1.

After this statement is executed, "maryd" can log into Adaptive Server. She is automatically treated as a "guest" user in master, with limited permissions, unless she has been specifically given access to master.

The following statement sets up a login account ("omar_khayyam") and password ("rubaiyat") for user and makes pubs2 the default database for this user:

```
sp_addlogin omar_khayyam, rubaiyat, pubs2
```

To specify a full name for a user and use the default database and language, you must specify null in place of the *defdb* and *deflanguage* parameters. For example:

```
sp_addlogin omar, rubaiyat, null, null,
    "Omar Khayyam"
```

Alternatively, you can specify a parameter name, in which case you do not have to specify all the parameters. For example:

```
sp_addlogin omar, rubaiyat,
    @fullname = "Omar Khayyam"
```

When you execute sp_addlogin, Adaptive Server adds a row to master.dbo.syslogins, assigns a unique server **user ID** (suid) for the new user, and fills in other information. When a user logs in, Adaptive Server looks in syslogins for the name and password provided by the user. The password column is encrypted with a one-way algorithm so it is not human-readable.

The suid column in syslogins uniquely identifies each user on Adaptive Server. A user's suid remains the same, no matter what database he or she is using. The suid 1 is always assigned to the default "sa" account that is created when Adaptive Server is installed. Other users' server user IDs are integers assigned consecutively by Adaptive Server each time sp_addlogin is executed.

Creating groups

Groups provide a convenient way to grant and revoke permissions to more than one user in a single statement. Groups enable you to provide a collective name to a group of users. They are especially useful if you administer an Adaptive Server installation that has a large numbers of users. Every user is a member of the group "public" and can also be a member of one other group. (Users remain in "public," even when they belong to another group.)

It is probably most convenient to create groups before adding users to a database, since sp_adduser can assign users to groups as well as add them to the database.

A System Administrator or the Database Owner can create a group at any time with sp_addgroup. The syntax is:

sp_addgroup *grpname*

The group name, a required parameter, must follow the rules for identifiers. The System Administrator can assign or reassign users to groups with sp_changegroup.

To set up the Senior Engineering group, use the following command while using the database to which you want to add the group:

```
sp_addgroup senioreng
```

sp_addgroup adds a row to sysusers in the current database. Therefore, each group in a database, as well as each user, has an entry in sysusers.

Adding users to databases

The Database Owner or a System Administrator can use sp_adduser to add a user to a specific database. The user must already have an Adaptive Server login. The syntax is:

sp_adduser *loginame* [, *name_in_db* [, *grpname*]]

where:

- *loginame* is the login name of an existing user.
- *name_in_db* specifies a name that is different from the login name by which the user is to be known inside the database.

  You can use this feature to accommodate users' preferences. For example, if there are five Adaptive Server users named Mary, each must have a different login name. Mary Doe might log in as "maryd", Mary Jones as "maryj", and so on. However, if these users do not use the same databases, each might prefer to be known simply as "mary" inside a particular database.

If no *name_in_db* parameter is given, the name inside the database is the same as loginame.

Note This capability is different from the alias mechanism described in "Using aliases in databases" on page 373, which maps the identity and permissions of one user to another.

- *grpname* is the name of an existing group in the database. If you do not specify a group name, the user is made a member of the default group "public." Users remain in "public" even if they are a member of another group. See "Changing a user's group membership" on page 371 for information about modifying a user's group membership.

sp_adduser adds a row to the sysusers system table in the current database. When a user has an entry in the sysusers table of a database, he or she:

- Can issue use *database_name* to access that database
- Will use that database by default, if the default database parameter was issued as part of sp_addlogin
- Can use sp_modifylogin to make that database the default

This example shows how a Database Owner could give access permission to "maryh" of the engineering group "eng," which already exists:

```
sp_adduser maryh, mary, eng
```

This example shows how to give "maryd" access to a database, keeping her name in the database the same as her login name:

```
sp_adduser maryd
```

This example shows how to add "maryj" to the existing "eng" group, keeping her name in the database the same as her login name by using null in place of a new user name:

```
sp_adduser maryj, null, eng
```

Users who have access to a database still need permissions to read data, modify data, and use certain commands. These permissions are granted with the grant and revoke commands, discussed in Chapter 11, "Managing User Permissions."

Adding a “guest” user to a database

Creating a user named “guest” in a database enables any user with an Adaptive Server account to access the database as a **guest** user. If a user issues the use *database_name* command, and his or her name is not found in the database’s sysusers or *sysalternates* table, Adaptive Server looks for a guest user. If there is one, the user is allowed to access the database, with the permissions of the guest user.

The Database Owner can add a guest entry to the sysusers table of the database with sp_adduser:

```
sp_adduser guest
```

The guest user can be removed with sp_dropuser, as discussed in “Dropping users” on page 365.

If you drop the guest user from the master database, server users who have not yet been added to any databases will be unable to log in to Adaptive Server.

Note Although more than one individual can be a guest user in a database, you can still use the user’s server user ID, which is unique within the server, to audit each user’s activity. For more information about auditing, see Chapter 12, “Auditing.”

“guest” user permissions

“Guest” inherits the privileges of “public.” The Database Owner and the owners of database objects can use grant and revoke to make the privileges of “guest” either more or less restrictive than those of “public.” See Chapter 11, “Managing User Permissions,” for a description of the “public” privileges.

When you install Adaptive Server, master..sysusers contains a guest entry. The installation script for the pubs2 database also contains a guest entry for its sysusers table.

“guest” user in user databases

In user databases, the Database Owner adds a guest user that permits all Adaptive Server users to use that database. This saves the owner from having to use sp_adduser to explicitly name each one as a database user.

You can use the guest mechanism to restrict access to database objects while allowing access to the database.

For example, the owner of the titles table could grant select permission on titles to all database users except "guest" by executing these commands:

```
grant select on titles to public
sp_adduser guest
revoke all on titles from guest
```

"guest" user in *pubs2* and *pubs3*

The "guest" user entry in the sample databases allows new Adaptive Server users to follow the examples in the *Transact-SQL User's Guide*. The guest is given a wide range of privileges, including:

- select permission and data modification permission on all of the user tables
- execute permission on all of the procedures
- create table, create view, create rule, create default, and create procedure permissions

Creating visitor accounts

The System Security Officer can use sp_addlogin to enter a login name and password that visiting users are instructed to use. Typically, such users are granted restricted permissions. A default database may be assigned.

Warning! A visitor user account is not the same as the "guest" user account. All users of the visitor account have the same server user ID; therefore, you cannot audit individual activity. Each "guest" user has a unique server ID, so you can audit individua activity and maintain individual accountability. Setting up a visitor account to be used by more tha one user is not recommended because you lose individual accountability.

You can add a vistor user account named “guest” to master..syslogins using sp_addlogin. This “guest” user account takes precedence over the system “guest” user account. Note that, if you add a visitor user named “guest” with sp_adduser, this impacts system databases such as sybsystemprocs and sybsystemdb, which are designed to work with system “guest” user in them.

Adding remote users

You can allow users on another Adaptive Server to execute stored procedures on your server by enabling remote access. Working with the System Administrator of the remote server, you can also allow users of your server to execute **remote procedure calls** to the remote server.

To enable remote procedure calls, both the local and the remote server must be configured. For information about setting up remote servers and adding remote users, see Chapter 13, “Managing Remote Servers.”

Note Remote users and remote procedure calls are not included in the evaluated configuration.

Number of user and login IDs

Adaptive Server supports over 2,000,000,000 logins per server and users per database. Adaptive Server uses negative numbers as well as positive numbers to increase the range of possible numbers available for IDs.

Limits and Ranges of ID Numbers

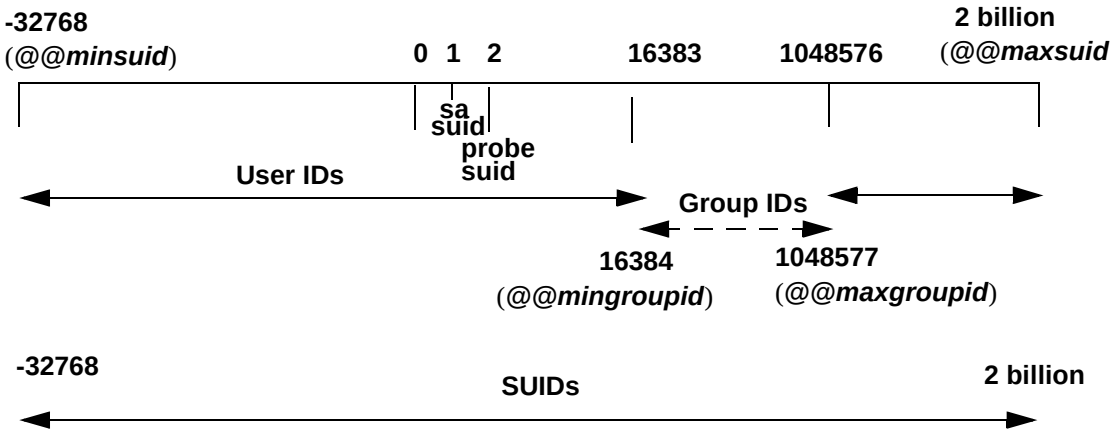
Table 10-2 describes the valid ranges for the ID types.

Table 10-2: Ranges for ID Types

ID Type	Server limits
Logins per server (*suid*)	2 billion plus 32K
Users per database (*uid*)	2 billion less 1032193
Groups per database (*guid*)	16,390 to 1,048,576

Figure 10-1 illustrates the limits and ranges for logins, users, and groups.

Figure 10-1: Users, groups, and logins available in Adaptive Server



Although Adaptive Server can handle over 2 billion users connecting at one time, the actual number of users that can connect to Adaptive Server is limited by:

- The number of user connections configuration parameter
- The number of file descriptors available from the operating system. Each user login uses one file descriptor per connection

Note Before Adaptive Server can have more than 64K logins and simultaneous connections, you must first configure the operating system for more than 64K file descriptors. See your operating system documentation for information about increasing the number of file descriptors

See the Release Bulletin for the most up to date information about Adaptive Server's limits for logins, users, and groups.

Login connection limitations

Although Adaptive Server allows you to define over 2,000,000,000 logins per server, the actual number of users that can connect to Adaptive Server at one time is limited by:

- The value of the number of user connections configuration parameter, and
- The number of file descriptors available for Adaptive Server. Each login uses one file descriptor for the connection.

Note The datatype of the server process ID (*spid*) has not been changed. Therefore, the maximum number of concurrent tasks running on the server is still thirty-two thousand.

To allow the maximum number of logins and simultaneous connections:

1 Configure the operating system on which Adaptive Server is running for at least thirty-two thousand file descriptors.

2 Set the value of number of user connections to at least thirty-two thousand.

Note Before Adaptive Server can have more than 64K logins and simultaneous connections, you must first configure the operating system for more than 64K file descriptors. See your operating system documentation for information about increasing the number of file descriptors.

Viewing Server Limits for Logins, Users, and Groups

Table 10-3 lists the global variables for the server limits of logins, users, and groups:

Table 10-3: Global variables for logins, users, and groups

Name of variable	What it displays	Value
@@invaliduserid	Invalid user ID	-1
@@minuserid	Lowest User ID	-32768
@@guestuserid	Guest user ID	2
@@mingroupid	Lowest group user ID	16384
@@maxgroupid	Highest group user ID	1048576
@@maxuserid	Highest user ID	2147483647
@@minsuid	Lowest server user ID	-32768
@@probesuid	Probe server user ID	2
@@maxsuid	Highest server user ID	2147483647

To issue a global variable, enter:

```
select variable_name
```

For example:

```
select @@minuserid
-----------
-32768
```

Creating and assigning roles to users

The final steps in adding database users are assigning them special roles, as required, and granting permissions. For more information on permissions, see Chapter 11, "Managing User Permissions."

The roles supported by Adaptive Server enable you to enforce individual accountability. Adaptive Server provides *system roles*, such as System Administrator and System Security Officer, and *user-defined roles*, which are created by a System Security Officer. Object owners can grant database access as appropriate to each role.

Table 10-4 lists the system roles, the value to use for the *role_granted* option of the grant role or revoke role command, and the tasks usually performed by a person with that role.

Table 10-4: System roles and related tasks

Role	Value for *role_granted*	Description
System Administrator	sa_role	Manages and maintains Adaptive Server databases and disk storage
System Security Officer	sso_role	Performs security-related tasks
Operator	oper_role	Backs up and loads databases server-wide

Note The sybase_ts_role, replication_role, and navigation_role roles are not included in the evaluated configuration.

Planning user-defined roles

Before you implement user-defined roles, decide:

- The roles you want to create
- The responsibilities for each role
- The position of each in the role hierarchy
- Which roles in the hierarchy will be mutually exclusive
- Whether such exclusivity will be at the membership level or activation level

User-defined role names cannot duplicate user names.

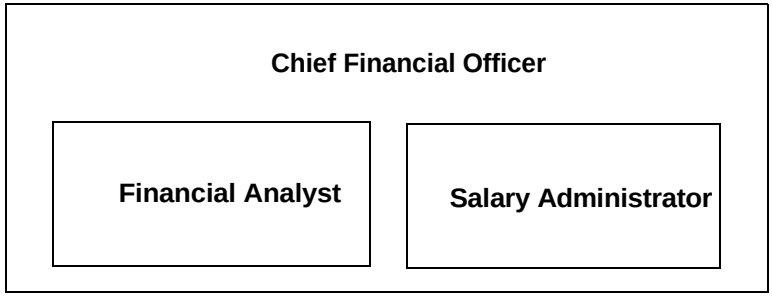
Avoid name-conflicts when you create user-defined roles by following a naming convention. For example, you could use the "_role" suffix for role names. Adaptive Server does not check for such restrictions.

If a role must have the same name as a user, you can avoid conflict by creating a new role, having it contain the original role, and then granting the new role to the user.

Role hierarchies and mutual exclusivity

A System Security Officer can define role hierarchies such that if a user has one role, the user also has roles lower in the hierarchy. For example, the "chief_financial_officer" role might contain both the "financial_analyst" and the "salary_administrator" roles, as shown in Figure 10-2.

Figure 10-2: Role hierarchy



The Chief Financial Officer can perform all tasks and see all data that can be viewed by the salary administrators and financial analysts.

Roles can be defined to be mutually exclusive for:

- Membership – One user cannot be granted two different roles. For example, you might not want the "payment_requestor" and "payment_approver" roles to be granted to the same user.
- Activation – One user cannot activate, or enable, two different roles. For example, a user might be granted both the "senior_auditor" and the "equipment_buyer" roles, but not permitted to have both roles enabled at the same time.

System roles, as well as user-defined roles, can be defined to be in a role hierarchy or to be mutually exclusive. For example, you might want a "super_user" role to contain the System Administrator, Operator, and "tech_support" roles. You might also want to define the System Administrator and System Security Officer roles to be mutually exclusive for membership; that is, one user cannot be granted both roles.

Configuring user-defined roles

After you have planned the roles to create and the relationships among them, configure your system for user-defined roles with the max roles enabled per user configuration parameter.

The maximum number of roles that a user can activate per user session is 127. The default value is 20. The minimum number of roles, which is 10, includes the system roles that come with Adaptive Server.

The maximum number of roles that can be activated server-wide is 992. The first 32 roles are reserved for Sybase system roles.

Creating a user-defined role

Use the create role command to create a role. The syntax is:

```
create role role_name [with passwd "password"]
```

where:

- *role_name* is the name of a new role.
- *password* is an optional password that must be specified by the user who will use the role.

For example, to create the intern_role without a password, enter:

```
create role intern_role
```

To create the doctor_role and assign the password “physician”, enter:

```
create role doctor_role with passwd "physician"
```

Adding and removing passwords from a role

Only a System Security Officer can add or drop a password from a role.

Use the alter role command to add or drop a password from either a system or user-defined role. The syntax is:

```
alter role role_name [add passwd password |
drop passwd]
```

For example, to require the password “oper8x” for the oper_role, enter:

```
alter role oper_role add passwd oper8x
```

To drop the password from the role, enter:

```
alter role oper_role drop passwd
```

Defining and changing mutual exclusivity of roles

To define mutual exclusivity between two roles, use:

alter role *role1* { add | drop } exclusive { membership | activation } *role2*

For example, to define intern_role and specialist_role as mutually exclusive at the membership level, enter:

```
alter role intern_role add exclusive membership
specialist_role
```

To define sso_role and sa_role as mutually exclusive at the activation level, enter:

```
alter role sso_role add exclusive activation sa_role
```

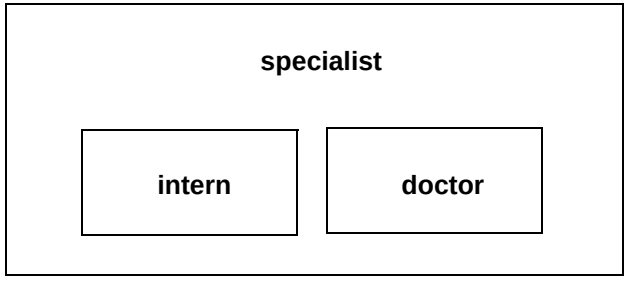
Defining and changing a role hierarchy

Defining a role hierarchy involves choosing the type of hierarchy and the roles, then implementing the hierarchy by granting roles to other roles.

For example:

```
grant role intern_role to specialist_role
grant role doctor_role to specialist_role
```

Figure 10-3: Creating a role hierarchy



In Figure 10-3, the “specialist” role contains the “doctor” and “intern” roles. This means that “specialist” has all the privileges of both “doctor” and “intern.”

To establish a hierarchy with a “super_user” role containing the sa_role and oper_role system roles, specify:

```
grant role sa_role to super_user
grant role oper_role to super_user
```

Note If a role that requires a password is contained within another role, the user with the role that contains the other does not need to use the password for the contained role. For example, in Figure 10-3, say the “doctor” role usually requires a password. The user who has the “specialist” role does not need to enter the “doctor” password because “doctor” is contained within “specialist.” Role passwords are only required for the highest level role.

When creating role hierarchies:

- You cannot grant a role to another role that directly contains it. This prevents duplication.

  For example, in Figure 10-3, you cannot grant “doctor” to “specialist” because “specialist” already contains “doctor.”

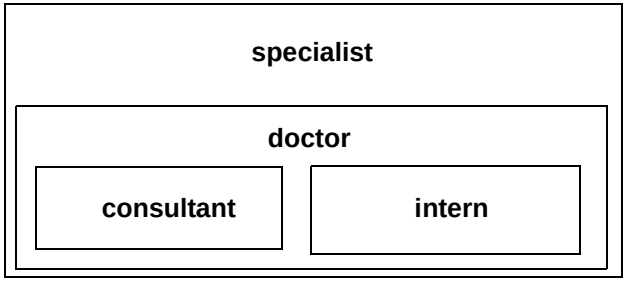
- You can grant a role to another role that does not directly contain it.

  For example, in Figure 10-4, you can grant the “intern” role to the “specialist” role, even though “specialist” already contains the “doctor” role, which contains “intern.” If you subsequently dropped “doctor” from “specialist,” then “specialist” would still contains “intern.”

  In Figure 10-4, “doctor” has “consultant” role permissions because “consultant” has been granted “doctor.” The “specialist” role also has “consultant” role permissions because “specialist” contains the “doctor” role, which in turn contains the “consultant.”

  However, “intern” does not have “consultant” role privileges, because “intern” does not contain the “consultant” role, either directly or indirectly.

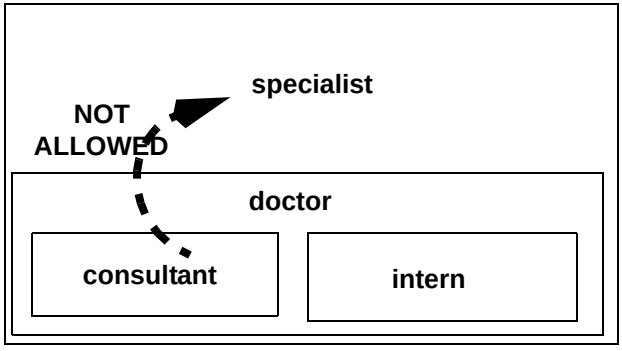
Figure 10-4: Explicitly and implicitly granted privileges



- You cannot grant a role to another role that is contained by the first role. This prevents "loops" within the hierarchy.

  For example, in Figure 10-5, you cannot grant the "specialist" role to the "consultant" role; "consultant" is already contained in "specialist".

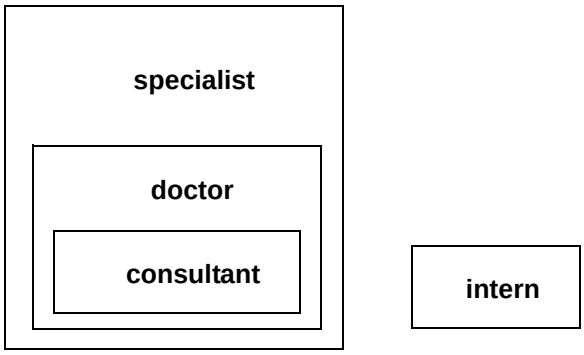
Figure 10-5: Granting a role to a role contained by grantor



- When the System Security Officer grants a user a role that contains other roles, the user implicitly gets membership in all roles contained by the granted role. However, a role can only be activated or deactivated directly if the user has explicit membership in that role.
- The System Security Officer cannot grant one role to another role that is explicitly or implicitly mutually exclusive at the membership level with the first role.

  For example, in Figure 10-6, if the "intern" role is defined as mutually exclusive at the membership level with the "consultant" role, the System Security Officer cannot grant "intern" to the "doctor."

Figure 10-6: Mutual exclusivity at membership

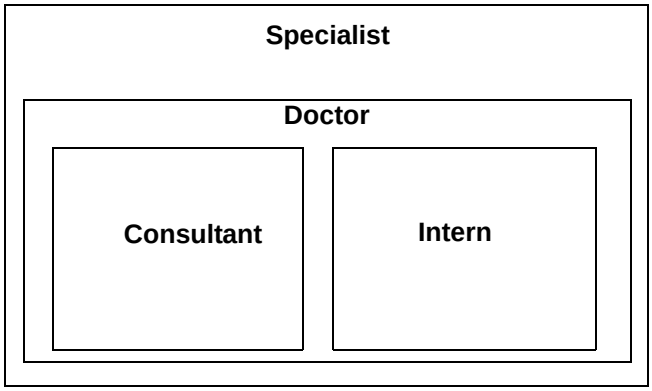


- The user can activate or deactivate only directly granted roles.

  For example, in the hierarchy shown in Figure 10-6, assume that you have been granted the “specialist” role. You have all the permissions of the “specialist” role, and, implicitly, because of the hierarchy, you have all the permissions of the “doctor” and “consultant” roles. However, you can activate only the “specialist” role. You cannot activate “doctor” or “consultant” because they were not directly granted to you. For information, see “Activating and deactivating roles” on page 364.

  Revoking roles from other roles is similar to granting roles to other roles. It removes a containment relationship, and the containment relationship must be a direct one, as shown in Figure 10-7:

Figure 10-7: Effect of revoking roles on role hierarchy



For example, in Figure 10-7:

- If the System Security Officer revokes the “doctor” role from “specialist,” “specialist” no longer contains the “consultant” role or the “intern” role.
- The System Security Officer cannot revoke the “intern” role from “specialist” because “intern” is not directly contained by “specialist.”

Setting up default activation at login

A System Security Officer can change a role’s default setting for any user. Individual users can change only their own default settings.

When a user logs in to Adaptive Server, the user’s roles are not necessarily active, depending upon the default that is set for the role. If a role has a password associated with it, the user must use the set role command to activate the role.

The System Security Officer or user determines whether to activate any roles granted by default at login. sp_modifylogin sets the default status of user roles individually for each user. sp_modifylogin only effects user roles, not system roles.

By default, user-defined roles are not activated at login, but system roles are automatically activated, if they do not have passwords associated with them.

To set up a role to activate at login:

sp_modifylogin *loginname*, "add default role", *role_name*

To ensure that a role is inactive at login:

sp_modifylogin *loginname*, "drop default role", *role_name*

For example, to change the default setting for Ralph's intern_role to be active automatically at login, execute:

sp_modifylogin ralph, "add default role", *intern_role*

Activating and deactivating roles

Roles must be active to have the access privileges of each role. Depending on the default set for a role, the role may or may not be active at login. If the role has a password, it will always be inactive at login.

To activate or deactivate a role:

set role role_name [on|off]

To activate or deactivate a role that has an attached password, use:

set role role_name with passwd "*password*" [on|off]

For example, to activate the "financial_analyst" role with the password "sailing19", enter:

```
set role financial_analyst with passwd "sailing19" on
```

You should activate roles only when you need them and turn them off when you no longer need them. For example, when the sa_role is active, you assume the identity of Database Owner within any database that you use. To turn off the System Administrator role and assume your "real" user identity, use:

```
set role sa_role off
```

If you are granted a role during a session, and you want to activate it immediately, use set role to turn it on.

Dropping users, groups and user-defined roles

Table 10-5 list the system procedures that allow a System Administrator or Database Owner to drop users and groups.

Table 10-5: Dropping users and groups

Task	Required Role	System Procedure	Database
Drop user from database	Database Owner or System Administrator	sp_dropuser	User database
Drop group from database	Database Owner or System Administrator	sp_dropgroup	User database

Dropping users

A Database Owner or a System Administrator can use sp_dropuser to deny an Adaptive Server user access to the database in which sp_dropuser is executed. (If a "guest" user is defined in that database, the user can still access that database as "guest.")

The syntax is:

```
sp_dropuser name_in_db
```

where *name_in_db* is usually the login name, unless another name has been assigned.

You cannot drop a user who owns objects. Since there is no command to transfer ownership of objects, you must drop objects owned by a user before you drop the user with sp_dropuser. To deny access to a user who owns objects, use sp_locklogin to lock his or her account.

You also cannot drop a user who has granted permissions to other users. Use revoke with cascade to revoke permissions from all users who were granted permissions by the user to be dropped, then drop the user. You must then grant permissions to the users again, if appropriate.

Dropping groups

Use sp_dropgroup to drop a group. The syntax is:

```
sp_dropgroup grpname
```

You cannot drop a group that has members. If you try to do so, the error report displays a list of the members of the group you are attempting to drop. To remove users from a group, use sp_changegroup, discussed in "Changing a user's group membership" on page 371.

Dropping user-defined roles

To drop a role, use:

drop role *role_name* [with override]

where *role_name* is the name of a user-defined role. with override revokes all access privileges granted to the role in every database server-wide.

If the role has any access privileges already granted, you must revoke all privileges granted to the role in all databases before you can drop the role. If you do not, the command fails. To revoke privileges:

- Use the revoke command, or
- Use the with override option with the drop role command. The with override option ensures that Adaptive Server automatically removes permission information for the role from all databases.

You need not drop memberships before dropping a role. Dropping a role automatically removes any user's membership in that role, regardless of whether you use the with override option.

Locking or dropping Adaptive Server login accounts

To prevent a user from logging in to Adaptive Server, you can either lock or drop an Adaptive Server login account. Locking a login is safer than dropping it because locking a login account maintains the suid so that it cannot be reused.

Warning! Adaptive Server may reuse the server user ID (suid) of a dropped login account when the next login account is created. This occurs only when the dropped login holds the highest suid in syslogins; however, it can compromise accountability if execution of sp_droplogin is not being audited. Also, it is possible for a user with the reused suid to access database objects that were authorized for the old suid.

You cannot drop a login when:

- The user is in any database
- The login belongs to the last remaining System Security Officer or System Administrator

Table 10-6: Locking or dropping login accounts

Task	Required role	System procedure	Database
Lock login account, which maintains the suid so that it cannot be reused	System Administrator or System Security Officer	sp_locklogin	master
Drop login account, which allows reuse of suid	System Administrator	sp_droplogin	master

Locking and unlocking login accounts

Use sp_locklogin to lock and unlock accounts or to display a list of locked accounts. You must be a System Administrator or a System Security Officer to use sp_locklogin.

The syntax is:

```
sp_locklogin [loginame, "{lock | unlock}"]
```

where:

- *loginame* is the name of the account to be locked or unlocked. It must be an existing valid account.
- lock | unlock specifies whether the account is to be locked or unlocked.

To display a list of all locked logins, use sp_locklogin with no parameters.

You can lock an account that is currently logged in, and the user is not locked out of the account until he or she logs out. You can lock the account of a Database Owner, and a locked account can own objects in databases. In addition, you can use sp_changedbowner to specify a locked account as the owner of a database.

Adaptive Server ensures that there is always at least one unlocked System Security Officer's account and one unlocked System Administrator's account.

Dropping login accounts

A System Administrator can use sp_droplogin to deny a user access to Adaptive Server. The syntax is:

```
sp_droplogin loginame
```

You cannot use sp_droplogin to drop a user from any database on the server. Use sp_dropuser to drop the user from a database. You cannot drop a user from a database if that user owns any objects in the database. For more information, see "Dropping users" on page 365.

Locking logins that own thresholds

This section discusses thresholds and how they are affected by locked user logins.

- As a security measure, threshold stored procedures are executed using the account name and roles of the login that created the procedure.
  - You cannot drop the login of a user who owns a threshold.
  - If you lock the login of a user who owns a threshold, the threshold cannot execute the stored procedure.
- Threshold procedures are executed with the most limited set of the roles assigned to the user. The user must have both of the following:
  - The set of roles active for the user at the time the threshold was added or last modified, and
  - The set of roles directly granted to the user at the time the threshold "fires."
- If a threshold requires a particular role, that role must be active for the user when the threshold is created. If that role is later revoked, the threshold cannot execute the procedure.
- The last chance threshold and thresholds created by the "sa" login are not affected by sp_locklogin. If you lock the "sa" login, the last chance threshold and thresholds created or modified by the "sa" user still fire.

Changing user information

Table 10-7 lists the system procedures you use to change passwords, default database, default language, full name, or group assignment.

Table 10-7: System procedures for changing user information

Task	Required role	System procedure	Database
Change your password	None	sp_password	Any database
Change another user's password	System Security Officer	sp_password	Any database
Change your default database, default language, or full name	None	sp_modifylogin	Any database
Change a login account's default database, default language, or full name	System Administrator	sp_modifylogin	Any database
Change the group assignment of a user	System Administrator, Database Owner	sp_changegroup	User database

Changing passwords

All users can change their passwords at any time using sp_password. The System Security Officer can use sp_password to change any user's password. The syntax is:

sp_password *caller_passwd*, *new_passwd* [, *loginame*]

where:

- *caller_passwd* is the password of the login account that is currently executing sp_password.
- *new_passwd* is the new password for the user executing sp_password, or for the user indicated by *loginame*. For guidelines on choosing and creating secure passwords, see "Choosing and creating a password" on page 344.
- *loginame* can be used only by a System Security Officer to change another user's password.

For example, a user can change his or her own password from "3blindmice" to "2mediumhot" using:

```
sp_password "3blindmice", "2mediumhot"
```

These passwords are enclosed in quotes because they begin with numbers.

In the following example, the System Security Officer whose password is "2tomato" changes Victoria's password to "sesame1":

```
sp_password "2tomato", sesame1, victoria
```

Requiring new passwords

Your site may choose to use the systemwide password expiration configuration parameter to establish a password expiration interval, which forces all Adaptive Server users to change their passwords on a regular basis. For information, see Chapter 5, "Setting Configuration Parameters." Even if you do not use systemwide password expiration, it is important, for security reasons, that users change their passwords periodically.

The column pwdate in the syslogins table records the date of the last password change. The following query selects all login names whose passwords have not changed since September 15, 1997:

```
select name, pwdate
from syslogins
where pwdate < "Sep 15 1997"
```

Null passwords

Do not assign a null password. When Adaptive Server is installed, the default "sa" account has a null password. The following example shows how to change a null password to a valid one:

```
sp_password null, "8M4LNCH"
```

Note that "null" is not enclosed in quotes in the statement.

Changing user defaults

Any user can use sp_modifylogin to change his or her default database, default language, or full name. sp_modifylogin only effects user roles, not system roles. A System Administrator can change these settings for any user. The syntax is:

sp_modifylogin *account*, *column*, *value*

- *account* is the name of the user whose account you are modifying.
- *column* specifies the option that you are changing. The options are:
  - defdb – The "home" database to which the user is connected when he or she logs in

- deflanguage – The official name of the user's default language, as stored in master..syslanguages
- fullname – The user's full name

- *value* is the new value for the specified option.

After you execute sp_modifylogin to change the default database, the user is connected to the new default database the next time he or she logs in. However, sp_modifylogin does not automatically give the user access to the database. Unless the Database Owner has set up access with sp_adduser, sp_addalias, or with a guest user mechanism, the user is connected to master even after his or her default database has been changed.

This example changes the default database for "anna" to pubs2:

```
sp_modifylogin anna, defdb, pubs2
```

This example changes the default language for "claire" to French:

```
sp_modifylogin claire, deflanguage, french
```

This example changes the full name for "mtwain" to "Samuel Clemens."

```
sp_modifylogin mtwain, fullname, "Samuel Clemens"
```

Changing a user's group membership

A System Administrator or the Database Owner can use sp_changegroup to change a user's group affiliation. Each user can be a member of only one group other than "public," of which all users are always members.

Before you execute sp_changegroup:

- The group must exist. (Use sp_addgroup to create a group.)
- The user must have access to the current database (must be listed in sysusers).

The syntax for sp_changegroup is:

sp_changegroup *grpname*, *username*

For example, to change the user "jim" from his current group to the group "manage," use:

```
sp_changegroup manage, jim
```

To remove a user from a group without assigning the user to another group, you must change the group affiliation to "public":

```
sp_changegroup "public", jim
```

The name “public” must be in quotes because it is a reserved word. This command reduces Jim’s group affiliation to “public” only.

When a user changes from one group to another, the user loses all permissions that he or she had as a result of belonging to the old group, but gains the permissions that have been granted to the new group.

The assignment of users into groups can be changed at any time.

Changing the user process information

The set command includes options that allow you to assign each client an individual name, host name, and application name. This is useful for differentiating among clients in a system where many clients connect to Adaptive Server using the same name, host name, or application name.

The partial syntax for the set command is:

set [clientname *client_name* | clienthostname *host_name* | clientapplname *application_name*]

Where *client_name* is the name you are assigning the client, *host_name* is the name of the host from which the client is connecting, and *application_name* is the application that is connecting to Adaptive Server. These parameters are stored in the clientname, clienthostname, clientapplname columns of the sysprocesses table.

For example, if a user logs in to Adaptive Server as “client1,” you can assign them an individual client name, host name, and application name using commands similar to:

```
set clientname 'alison'
set clienthostname 'money1'
set clientapplname 'webserver2'
```

This user now appears in the sysprocesses table as user “alison” logging in from host “money1” and using the “webserver2” application. However, although the new names appear in sysprocesses, they are not used for permission checks, and sp_who still shows the client connection as belonging to the original login (in the case above, client1). set clientname does not perform the same function as set proxy, which allows you to assume the permissions, login name, and suid of another user.

You can set a client name, host name, or application name for only your current client session (although you can view the connection information for any client connection). Also, this information is lost when a user logs out. These parameters must be reassigned each time a user logs in. For example, the user alison cannot set the client name, host name, or application name for any other client connection.

Use the client's spid to view their connection information. For example, if the client "alison" described above connects with a spid of 13, issue the following command to view all the connection information for this client:

```
select * from sysprocesses where spid = 13
```

To view the connection information for the current client connection (for example, if the user alison wanted to view her own connection information), enter:

```
select * from sysprocesses where spid = @@spid
```

Using aliases in databases

The alias mechanism allows you to treat two or more users as the same user inside a database so that they all have the same privileges. This mechanism is often used so that more than one user can assume the role of Database Owner. A Database Owner can use the setuser command to impersonate another user in the database. You can also use the alias mechanism to set up a collective user identity.

For example, suppose that several vice presidents want to use a database with identical privileges and ownerships. If you add the login "vp" to Adaptive Server and the database and have each vice president log in as "vp," there is no way to tell the individual users apart. Instead, alias all the vice presidents, each of whom has his or her own Adaptive Server account, to the database user name "vp."

Note Although more than one individual can use the alias in a database, you can still maintain individual accountability by auditing the database operations performed by each user. For more information about auditing, see Chapter 12, "Auditing."

Table 10-8 lists the system procedures used to manage aliases:

Table 10-8: System procedures for managing aliases

Task	Require role	System procedure	Database
Add an alias for a user	Database Owner or System Administrator	sp_addalias	User database
Drop an alias	Database Owner or System Administrator	sp_dropalias	User database

Note As of version 12.0, you cannot drop the alias of a login if that login created objects in the database. In most cases, you should use aliases only for users who do not own tables, procedures, views or triggers.

Adding aliases

To add an alias for a user, use sp_addalias. The syntax is:

```
sp_addalias loginame, name_in_db
```

where:

loginame is the name of the user who wants an alias in the current database. This user must have an account in Adaptive Server but cannot be a user in the current database.

name_in_db is the name of the database user to whom the user specified by *loginame* is to be linked. The *name_in_db* must exist in both master..syslogins and in sysusers in the current database.

Executing sp_addalias maps the user name specified by *loginame* to the user name specified by *name_in_db*. It does this by adding a row to the system table sysalternates.

When a user tries to use a database, Adaptive Server checks for the user's server user ID number (suid) in sysusers. If it is not found, Adaptive Server then checks sysalternates. If the user's *suid* is found there, and it is mapped to a database user's *suid*, the first user is treated as the second user while the first user is using the database.

For example, suppose that Mary owns a database. She wants to allow both Jane and Sarah to use the database as if they were its owner. Jane and Sarah have logins on Adaptive Server but are not authorized to use Mary's database. Mary executes the following commands:

```
sp_addalias jane, dbo
exec sp_addalias sarah, dbo
```

Warning! Users who are aliased to the Database Owner have all the permissions and can perform all the actions that can be performed by the real Database Owner, with respect to the database in question. A Database Owner should carefully consider the implications of vesting another user with full access to a database.

Dropping aliases

Use sp_dropalias to drop the mapping of an alternate *suid* to a user ID. Doing this deletes the relevant row from sysalternates. The syntax is:

sp_dropalias *loginame*

where *loginame* is the name of the user specified by *loginame* when the name was mapped with sp_addalias. After a user's alias is dropped, the user no longer has access to the database.

You cannot drop an alias for a user who owns objects in the database that were created with version 12.0 or later. You must drop the objects (re-creating them under a different login, if needed) before you can drop the alias.

Getting information about aliases

To display information about aliases, use sp_helpuser. For example, to find the aliases for "dbo," execute:

```
sp_helpuser dbo
Users_name     ID_in_db    Group_name   Login_name
----------     --------    ----------   ----------
dbo            1           public       sa

(1 row affected)
                           Users aliased to user.
                           Login_name
                           ----------------------
                           andy
                           christa
                           howard
```

```
linda

(4 rows affected)
```

Getting information about users

Table 10-9 lists procedures you can use to obtain information about users, groups, and current Adaptive Server usage.

Table 10-9: Reporting information about Adaptive Server users and groups

Task	Procedure
Report current Adaptive Server users and processes	sp_who
Display information about login accounts	sp_displaylogin
Report users and aliases in a database	sp_helpuser
Report groups within a database	sp_helpgroup

Getting reports on users and processes

Use sp_who to report information about current users and processes on Adaptive Server:

sp_who [*loginame* | "*spid*"]

where:

- *loginame* is the user's Adaptive Server login name. If you give a login name, sp_who reports information about processes being run by that user.
- *spid* is the number of a specific process.

For each process being run, sp_who reports the server process ID, its status, the login name of the process user, the name of the host computer, the server process ID of a process that's blocking this one (if any), the name of the database, and the command being run.

If you do not give a login name or *spid*, sp_who reports on processes being run by all users.

The following example shows the results of executing sp_who without a parameter:

```
spid   status   loginame hostname blk dbname cmd
------ -------- -------- -------- --- ------ ----------------
     1 running  sa       sunbird  0   pubs2  SELECT
     2 sleeping NULL              0   master NETWORK HANDLER
     3 sleeping NULL              0   master MIRROR HANDLER
     4 sleeping NULL              0   master AUDIT PROCESS
     5 sleeping NULL              0   master CHECKPOINT SLEEP

(5 rows affected, return status = 0)
```

sp_who reports NULL for the *loginame* for all system processes

Getting information about login accounts

Use sp_displaylogin to display information about a specified login account, including any roles granted to that account:

sp_displaylogin [*loginame*]

where *loginame* is the user login account about which you want information. If you are not a System Security Officer or System Administrator, you can get information only about your own account. If you are a System Security Officer or System Administrator, you can use the *loginame* parameter to access information about any account.

sp_displaylogin displays your server user ID, login name, full name, any roles that have been granted to you, date of last password change, default database, default language, and whether your account is locked.

sp_displaylogin displays all roles that have been granted to you, so even if you have made a role inactive with the set command, that role is displayed.

Getting information about database users

Use sp_helpuser to report information about authorized users of the current database:

sp_helpuser [*name_in_db*]

where *name_in_db* is the user's name in the current database. If you give a user's name, sp_helpuser reports information about that user. If you do not give a name, it reports information about all users.

The following example shows the results of executing sp_helpuser without a parameter in the database pubs2:

```
sp_helpuser
Users_name  ID_in_db  Group_name Login_name
----------  --------  ---------- ----------
dbo         1         public     sa
marcy       4         public     marcy
sandy       3         public     sandy
judy        5         public     judy
linda       6         public     linda
anne        2         public     anne
jim         7         senioreng  jim
```

Finding user names and IDs

To find a user's server user ID or login name, use suser_id and suser_name.

Table 10-10: System functions suser_id and suser_name

To find	Use	With the argument
Server user ID	suser_id	(["server_user_name"])
Server user name (login name)	suser_name	([server_user_ID])

The arguments for these system functions are optional. If you do not provide one, Adaptive Server displays information about the current user.

This example shows how to find the server user ID for the user "sandy:"

```
select suser_id("sandy")
------
     3
```

This example shows how a System Administrator whose login name is "mary" issues the commands without arguments:

```
select suser_name(), suser_id()
------------------------------ ------
mary                                4
```

To find a user's ID number or name inside a database, use user_id and user_name.

Table 10-11: System functions user_id and user_name

To find	Use	With the argument
User ID	user_id	(["db_user_name"])
User name	user_name	([db_user_ID])

The arguments for these functions are optional. If you do not provide one, Adaptive Server displays information about the current user. For example:

```
select user_name(10)
select user_name( )
select user_id("joe")
```

Displaying information about roles

Table 10-12 lists the system procedures and functions to use to find information about roles and the section in this Fchapter that provides details.

Table 10-12: Finding information about roles

To Display Information About	Use	See
The role ID of a role name	role_id system function	"Finding role IDs and names" on page 380
The role name of a role ID	role_name system function	"Finding role IDs and names" on page 380
System roles	show_role system function	"Viewing active roles" on page 380
Role hierarchies and roles that have been granted to a user or users	sp_displayroles system procedure	"Displaying a role hierarchy" on page 380
Whether one role contains another role in a role hierarchy	role_contain system function	"Viewing user roles in a hierarchy" on page 381
Whether two roles are mutually exclusive	mut_excl_roles system function	"Determining mutual exclusivity" on page 381
Roles that are active for the current session	sp_activerolessystem procedure	"Determining role activation" on page 381
Whether you have activated the correct role to execute a procedure	proc_role system function	"Checking for roles in stored procedures" on page 381
Logins, including roles that have been granted	sp_displaylogin system procedure	"Getting information about login accounts" on page 377
Permissions for a user, group, or role	sp_helprotect system procedure	"Reporting on permissions" on page 426

Finding role IDs and names

To find a role ID when you know the role name, use role_id:

```
role_id(role_name)
```

Any user can execute role_id. If the role is valid, role_id returns the server-wide ID of the role (srid). The syssrvroles system table contains an srid column with the role ID and a name column with the role name. If the role is not valid, role_id returns NULL.

To find a role name when you know the role ID, use role_name:

```
role_name(role_id)
```

Any user can execute role_name.

Viewing active roles

Use show_role to display the currently active *system roles* for the specified login:

```
show_role()
```

If you have not activated any system role, show_role returns NULL. If you are a Database Owner, and you execute show_role after using setuser to impersonate another user, show_role returns your own active system roles, not those for whom you are impersonating.

Any user can execute show_role.

Note The show_role function does not give information about user-defined roles.

Displaying a role hierarchy

You can see all roles granted to your login name or see the entire hierarchy tree of roles displayed in table format using sp_displayroles:

```
sp_displayroles {login_name | rolename [, expand_up |
expand_down]}
```

Any user can execute sp_displayroles to see his or her own roles. Only the System Security Officer or the System Administrator can view information about roles granted to other users.

Viewing user roles in a hierarchy

Use role_contain to determine whether any role you specify contains any other role you specify:

```
role_contain (["role1", "role2"])
```

If *role1* contains *role2*, role_contain returns 1.

Any user can execute role_contain.

Determining mutual exclusivity

Use the mut_excl_roles function to determine whether any two roles assigned to you are mutually exclusive and the level at which they are mutually exclusive:

```
mut_excl(role1, role2, {membership | activation})
```

Any user can execute mut_excl_roles. If the specified roles, or any role contained by either specified role, are mutually exclusive, mut_excl_roles returns 1; if the roles are not mutually exclusive, mut_excl_roles returns 0.

Determining role activation

To find all active roles for the current login session of Adaptive Server, use sp_activeroles:

```
sp_activeroles [expand_down]
```

expand_down displays the hierarchy of all roles contained by any roles granted to you.

Any user can execute sp_activeroles.

Checking for roles in stored procedures

Use proc_role within a stored procedure to guarantee that only users with a specific role can execute the procedure. Only proc_role provides a fail-safe way to prevent inappropriate access to a particular stored procedure.

You can use grant execute to grant execute permission on a stored procedure to all users who have been granted a specified role. Similarly, revoke execute removes this permission.

However, grant execute permission does not prevent users who do not have the specified role from being granted execute permission on a stored procedure. If you want to ensure, for example, that all users who are not System Administrators can never be granted permission to execute a stored procedure, use proc_role within the stored procedure itself. It checks to see whether the invoking user has the correct role to execute the procedure.

proc_role takes a string for the required role and returns 1 if the invoker possesses it. Otherwise, it returns 0.

For example, here is a procedure that uses proc_role to see if the user has the sa_role role:

```
create proc test_proc
as
if (proc_role("sa_role") = 0)
begin
    print "You don’t have the right role"
    return -1
end
else
    print "You have System Administrator role"
    return 0
```

Monitoring license use

The License Use Monitor allows a System Administrator to monitor the number of user licenses used in Adaptive Server and securely manage the license agreement data. That is, you can ensure that the number of licenses used on your Adaptive Server does not exceed the number specified in your license agreement.

The License Use Monitor tracks the number of licenses issued; it does not enforce the license agreement. If the License Use Monitor reports that you are using more user licenses than specified in your license agreement, see your Sybase sales representative.

You must have System Administrator privileges to configure the License Use Monitor.

By default, the License Use Monitor is turned off when Adaptive Server is first installed or upgraded. The System Administrator must configure the License Use Monitor to monitor license usage. See "“Configuring License Manager to monitor user licenses” on page 383 for configuration information.

How licenses are counted

A license is the combination of a host computer name and a user name. If a user logs in to Adaptive Server multiple times from the same host machine, it counts as one license. However, if the user logs in once from host A, and once from host B, it counts as two licenses. If multiple users log in to Adaptive Server from the same host, but with different user names, each distinct combination of user name and host name is counted.

Configuring License Manager to monitor user licenses

Use sp_configure to specify the number of licenses in your license agreement:

```
sp_configure "license information" , number
```

where *number* is the number of licenses. For example:

```
sp_configure "license information", 300
```

sets the maximum number of user licenses to 300, and reports an overuse for license number 301. If you increase the number of user licenses, you must also change the license number configuration parameter.

The configuration parameter housekeeper free write percent must be set to 1 or more in order for the License Manager to track license use.

Monitoring license use with the housekeeper task

After you configure the License Use Monitor, the housekeeper task determines how many user licenses are in use, based on the user ID and the host name of each user logged in to Adaptive Server. When the housekeeper task checks licenses, the License Use Monitor updates a variable that tracks the maximum number of user licenses in use:

- If the number of licenses in use is the same or has decreased since the previous housekeeper run, the License Use Monitor does nothing
- If the number of licenses in use has increased since the previous housekeeper run, the License Use Monitor sets this number as the maximum number of licenses in use.
- If the number of licenses in use is greater than the number allowed by the license agreement, the License Use Monitor issues message to the error log:

```
Exceeded license usage limit. Contact Sybase
Sales for additional licenses.
```

The housekeeper task runs during Adaptive Server's idle cycles. The housekeeper monitors the number of user licenses only if the housekeeper free write percent configuration parameter is set to 1 or greater.

For more information about the housekeeper task, see "housekeeper free write percent" on page 187 and Chapter 3, "Using Engines and CPUs," in the *Performance and Tuning Guide*.

Logging the number of user licenses

The *syblicenseslog* system table is created in the master database when you install or upgrade Adaptive Server. The License Use Monitor updates the columns in syblicenseslog at the end of each 24-hour period, as shown in Table 10-13.

Table 10-13: Columns in syblicenseslog table

Column	Description
status	-1 – Housekeeper unable to monitor licenses. 0 – Number of licenses not exceeded. 1 – Number of licensees exceeded.
logtime	Date and time the log information was inserted.
maxlicenses	Maximum number of licenses used during the previous 24 hours.

syblicenseslog looks similar to this:

```
status logdate                    maxlicenses
 ------ -------------------------- -----------
      0         Jul 17 1998 11:43AM         123
      0         Jul 18 1998 11:47AM         147
      1         Jul 19 1998 11:51AM         154
```

```
0          Jul 20 1998 11:55AM          142
0          Jul 21 1998 11:58AM          138
0          Jul 21 1998  3:14PM          133
```

In this example, the number of user licenses used exceeded the limit on July 19, 1998.

If Adaptive Server is shut down, License Manager updates syblicenseslog with the current maximum number of licenses used. Adaptive Server starts a new 24-hour monitoring period when it is rebooted.

The second row for July 21, 1998 was caused by a shutdown and reboot of the server.

Getting information about usage: Chargeback accounting

When a user logs in to Adaptive Server, the server begins accumulating CPU and I/O usage for that user. Adaptive Server can report total usage for an individual or for all users. Information for each user is kept in the syslogins system table in the master database.

Reporting current usage statistics

The System Administrator can use sp_reportstats or sp_clearstats to get or clear current total usage data for individuals or for all users on Adaptive Server.

Displaying current accounting totals

sp_reportstats displays current accounting totals for Adaptive Server users. It reports total CPU and total I/O, as well as the percentage of those resources used. It does not record statistics for the “sa” login (processes with an *suid* of 1), checkpoint, network, and mirror handlers.

Initiating a new accounting interval

Adaptive Server accumulates CPU and I/O statistics until you clear the totals from syslogins by running sp_clearstats. sp_clearstats initiates a new accounting interval for Adaptive Server users and executes sp_reportstats to print out statistics for the previous period.

Choose the length of your accounting interval by deciding how you want to use the statistics at your site. For example, to do monthly cross-department charging for the percentage of Adaptive Server CPU and I/O usage, the System Administrator would run sp_clearstats once a month.

For detailed information about these stored procedures, see the *Adaptive Server Reference Manual*.

Specifying the interval for adding accounting statistics

A System Administrator can use configuration parameters to decide how often accounting statistics are added to syslogins.

To specify how many machine clock ticks accumulate before accounting statistics are added to syslogins, use the cpu accounting flush interval configuration parameter. The default value is 200. For example:

```
sp_configure "cpu accounting flush interval", 600
```

To find out how many microseconds a tick is on your system, run the following query in Adaptive Server:

```
select @@timeticks
```

To specify how many read or write I/Os accumulate before the information is added (flushed) to syslogins, use the i/o accounting flush interval configuration parameter. The default value is 1000. For example:

```
sp_configure "i/o accounting flush interval", 2000
```

I/O and CPU statistics are flushed when a user accumulates more I/O or CPU usage than the specified value. The information is also flushed when the user exits an Adaptive Server session.

The minimum value allowed for either configuration parameter is 1. The maximum value allowed is 2,147,483,647.

CHAPTER 11

Managing User Permissions

This chapter describes the use and implementation of user permissions.

Topics covered in this chapter include:

Overview

Discretionary access controls (DAC) allow you to restrict access to objects and commands based on a user's identity or group membership. The controls are "discretionary" because a user with a certain access permission, such as an object owner, can choose whether to pass that access permission on to other users.

System Administrators operate outside the DAC system and have access permissions on all database objects at all times. System Security Officers can always access the audit trail tables in the sybsecurity database.

Database Owners do not automatically receive permissions on objects owned by other users; however, they can:

- Temporarily acquire all permissions of a user in the database by using the setuser command to assume the identity of that user.
- Permanently acquire permission on a specific object by using the setuser command to assume the identity of the object owner, and then using grant commands to grant the permissions.

For details on assuming another user's identity to acquire permissions on a database or object, see "Acquiring the permissions of another user" on page 420.

Object Owners can grant access to those objects to other users and can also grant other users the ability to pass the access permission to other users. You can give various permissions to users, groups, and roles with the grant command, and rescind them with the revoke command. Use these commands to give users permission to create databases, to create objects within a database, execute certain commands such as set proxy, and to access specified tables, views, and columns. For permissions that default to "public," no grant or revoke statements are needed.

Some commands can be used at any time by any user, with no permission required. Others can be used only by users of a particular status and they are not transferable.

The ability to assign permissions for the commands that can be granted and revoked is determined by each user's role or status (as System Administrator, Database Owner, or database object owner), and by whether the user was granted a role with permission that includes the option to grant that permission to other users.

You can also use views and stored procedures as security mechanisms. See "Using views and stored procedures as security mechanisms" on page 431.

Types of users and their privileges

Adaptive Server's discretionary access control system recognizes the following types of users:

- System Administrators
- System Security Officers
- Operators
- Database Owners
- Database object owners
- Other users (also known as "public")

System Administrator privileges

System Administrators:

- Handle tasks that are not specific to applications
- Work outside Adaptive Server's discretionary access control system

The role of System Administrator is usually granted to individual Adaptive Server logins. All actions taken by that user can be traced to his or her individual server user ID. If the server administration tasks at your site are performed by a single individual, you may instead choose to use the "sa" account that is installed with Adaptive Server. At installation, the "sa" account user has permission to assume the System Administrator, System Security Officer, and Operator roles. Any user who knows the "sa" password can log in to that account and assume any or all of these roles.

The fact that a System Administrator operates outside the protection system serves as a safety precaution. For example, if the Database Owner accidentally deletes all the entries in the sysusers table, the System Administrator can restore the table (as long as backups exist). There are several commands that can be issued only by a System Administrator. They include disk init, disk refit, disk reinit, shutdown, kill, and the disk mirroring commands.

In granting permissions, a System Administrator is treated as the object owner. If a System Administrator grants permission on another user's object, the owner's name appears as the grantor in sysprotects and in sp_helprotect output.

In addition, System Administrators are responsible for dropping logins and can lock and unlock logins. System Security Officers share login management responsibilities with System Administrators. System Security Officers are responsible for adding logins and can also lock and unlock logins.

Permissions for creating databases

Only a System Administrator can grant permission to use the create database command. The user that receives create database permission must also be a valid user of the master database because all databases are created while using master.

In many installations, the System Administrator maintains a monopoly on create database permission to centralize control of database placement and database device space allocation. In these situations, a System Administrator creates new databases on behalf of other users, and then transfers ownership to the appropriate user.

To create a database that is to be owned by another user:

1 Issue the create database command in the master database.

2 Switch to the new database with the use command.

3 Execute sp_changedbowner.

System Security Officer privileges

System Security Officers perform security-sensitive tasks in Adaptive Server, including:

- Granting the System Security Officer and Operator roles
- Administering the audit system
- Changing passwords
- Adding new logins
- Locking and unlocking login accounts
- Creating and granting user-defined roles
- Administering network-based security
- Granting permission to use the set proxy or set session authorization commands

The System Security Officer can *access* any database – to enable auditing – but, in general, has no special permissions on database objects. An exception is the sybsecurity database, where only a System Security Officer can access the sysaudits table. There are also several system procedures that can be executed only by a System Security Officer.

System Security Officers can repair any damage inadvertently done to the protection system by a user. For example, if the Database Owner forgets his or her password, a System Security Officer can change the password to allow the Database Owner to log in.

System Security Officers can also create and grant user-defined roles to users, other roles, or groups. For information about creating and granting user-defined roles, see Chapter , "Creating and assigning roles to users."

Operator privileges

Users who have been granted the Operator role can back up and restore databases on a server-wide basis without having to be the owner of each database. The Operator role allows a user to use these commands on any database:

dump database
dump transaction
load database
load transaction

Database Owner privileges

Database Owners and System Administrators are the only users who can grant object creation permissions to other users. The Database Owner has full privileges to do anything inside that database, and must explicitly grant permissions to other users with the grant command.

Permission to use these commands is automatically granted to the Database Owner and cannot be transferred to other users:

checkpoint
dbcc
drop database
dump database
dump transaction
grant (object creation permissions)
load database
load transaction
revoke (object creation permissions)
setuser

Database Owners can grant permission to use these commands to other users:

create default
create procedure
create rule
create table
create view
grant (permissions on system tables)
grant (select, insert, delete, update, references, and execute permissions on database objects)
revoke (permissions on system tables)
revoke (select, insert, delete, update, references, and execute permissions on database objects)

Permissions on system tables

Permissions for use of the system tables can be controlled by the Database Owner, just like permissions on any other tables. By default, when Adaptive Server is installed, the installmodel script grants select access to "public" (all users) for most system tables and for most fields in the tables. However, no access is given for some system tables, such as systhresholds, and no access is given for certain fields in other system tables. For example, all users, by default, can select all columns of sysobjects except audflags.

To determine the current permissions for a particular system table, execute:

sp_helprotect *system_table_name*

For example, to check the permissions of systhresholds in your_database, execute:

```
use your_database
go
sp_helprotect systhresholds
go
```

The default situation is that no users—including Database Owners—can modify the system tables directly. Instead, the system procedures supplied with Adaptive Server modify the system tables. This helps guarantee integrity.

Warning! Although does provide a mechanism that allows you to modify system tables, Sybase strongly recommends that you do not do so.

Permissions on system procedures

Permissions on system procedures are set in the sybsystemprocs database, where the system procedures are stored.

Security-related system procedures can be run only by System Security Officers. Certain other system procedures can be run only by System Administrators.

Some of the system procedures can be run only by Database Owners. These procedures make sure that the user executing the procedure is the owner of the database from which they are being executed.

Other system procedures can be executed by any user who has been granted permission. A user must have permission to execute a system procedure in all databases, or in none of them.

Users who are not listed in sybsystemprocs..sysusers are treated as "guest" in sybsystemprocs, and are automatically granted permission on many of the system procedures. To deny a user permission on a system procedure, the System Administrator must add him or her to sybsystemprocs..sysusers and issue a revoke statement that applies to that procedure. The owner of a user database cannot directly control permissions on the system procedures from within his or her own database.

Changing database ownership

Use sp_changedbowner to change the ownership of a database. Often, System Administrators create the user databases, then give ownership to another user after some of the initial work is complete. Only the System Administrator can execute sp_changedbowner.

It is a good idea to transfer ownership before the user has been added to the database, and before the user has begun creating objects in the database. The new owner must already have a login name on Adaptive Server, but cannot be a user of the database, or have an alias in the database. You may have to use sp_dropuser or sp_dropalias before you can change a database's ownership, and you may have to drop objects before you can drop the user.

Issue sp_changedbowner in the database whose ownership will be changed. The syntax is:

```
sp_changedbowner loginame [, true ]
```

This example makes “albert” the owner of the current database and drops aliases of users who could act as the old “dbo:”

```
sp_changedbowner albert
```

To transfer aliases and their permissions to the new “dbo,” add the value true parameters.

Note You cannot change the ownership of the master database and should not change the ownership of any other system databases.

Database object owner privileges

A user who creates a database object (a table, view, or stored procedure) owns the object and is automatically granted all object access permissions on it. Users other than the object owner, including the owner of the database, are automatically denied all permissions on that object, unless they are explicitly granted by either the owner or a user who has grant permission on that object.

As an example, suppose that Mary is the owner of the pubs2 database, and has granted Joe permission to create tables in it. Now Joe creates the table new_authors; he is the owner of this database object.

Initially, object access permissions on new_authors belong only to Joe. Joe can grant or revoke object access permissions for this table to other users.

The following object creation permissions default to the owner of a table and cannot be transferred to other users:

alter table
drop table
create index
truncate table
update statistics

Permission to use the grant and revoke commands to grant specific users select, insert, update, delete, references, and execute permissions on specific database objects can be transferred, using the grant with grant option command.

Permission to drop an object—a table, view, index, stored procedure, rule, or default—defaults to the object owner and cannot be transferred.

Privileges of other database users

At the bottom of the hierarchy are other database users. Permissions are granted to or revoked from them by object owners, Database Owners, users who were granted permissions, or a System Administrator. These users are specified by user name, group name, or the keyword public.

Granting and revoking permissions on database objects

Two types of permissions exist for objects:

- **Object access permissions** – For using the commands that access database objects. For more information, see "Granting and revoking object access permissions" on page 395.
- **Object creation permissions** – For creating objects. They can be granted only by a System Administrator or a Database Owner. For more information, see "Granting and revoking object creation permissions" on page 401.

Both types of permissions are controlled with the grant and revoke commands.

Each database has its own independent protection system. Having permission to use a certain command in one database does not give you permission to use that command in other databases.

Granting and revoking object access permissions

Object access permissions regulate the use of certain commands that access certain database objects. For example, you must explicitly be granted permission to use the select command on the authors table. Object access permissions are granted and revoked by the object owner (and System Administrators), who can grant them to other users.

Table 11-1 lists the types of object access permissions and the objects to which they apply.

Table 11-1: Permissions and the objects to which they apply

Permission	Object
select	Table, view, column
update	Table, view, column
insert	Table, view
delete	Table, view
references	Table, column
execute	Stored procedure

The references permission refers to referential integrity constraints that you can specify in an alter table or create table command. The other permissions refer to SQL commands. Object access permissions default to System Administrators and the object's owner, and can be granted to other users.

Use the grant command to grant object access permissions. The syntax is:

```
grant {all [privileges]| permission_list}
    on { table_name [(column_list)]
        | view_name[(column_list)]
        | stored_procedure_name}
    to {public | name_list | role_name}
    [with grant option]
```

Use the revoke command to revoke object access permissions. The syntax is:

```
revoke [grant option for]
    {all [privileges] | permission_list}
    on { table_name [(column_list)]
        | view_name [(column_list)]
        | stored_procedure_name}
    from {public | name_list | role_name}
    [cascade]
```

Notes on the keywords and parameters are as follows:

- all or all privileges specifies all permissions applicable to the specified object. All object owners can use all with an object name to grant or revoke permissions on their own objects. If you are granting or revoking permissions on a stored procedure, all is the same as execute.

  Note insert and delete permissions do not apply to columns, so you cannot include them in a permission list (or use the keyword all) if you specify a column list.

- *permission_list* is the list of permissions that you are granting. If you name more than one permission, separate them with commas. Table 11-2 illustrates the access permissions that can be granted on each type of object:

Table 11-2: Object access permissions

Object	permission_list Can include
Table or view	select, insert, delete, update, references. references applies to tables but not views; the other permissions apply to both tables and views.
Column	select, update, references
Stored procedure	execute

  You can specify columns in the *permission_list* or the *column_list*, but not both.

- on specifies the object for which the permission is being granted or revoked. You can grant or revoke permissions for only one table, view, or stored procedure object at a time. You can grant or revoke permissions for more than one column at a time, but all the columns must be in the same table or view. You can only grant or revoke permissions on objects in your current database.
- public refers to the group "public," which includes all Adaptive Server users. public means slightly different things for grant and revoke:
  - For grant, public includes the object owner. Therefore, if you have revoked permissions from yourself on your object, and later you grant permissions to public, you regain the permissions along with the rest of "public."
  - For revoke, public excludes the owner.
- name_list includes:

  - Group names
  - User names
  - A combination of user and group names, each separated from the next by a comma

- *role_name* is an Adaptive Server system-defined or user-defined role. You can create and define a hierarchy of user-defined roles and grant them privileges based on the specific role granted. System-defined roles include sa_role (System Administrator), sso_role (System Security Officer), and oper_role (Operator). You cannot create or modify system-defined roles.
- with grant option in a grant statement allows the user(s) specified in *name_list* to grant the specified object access permission(s) to other users. If a user has with grant option permission on an object, that permission is not revoked when permissions on the object are revoked from public or a group of which the user is a member.
- grant option for revokes with grant option permissions, so that the user(s) specified in *name_list* can no longer grant the specified permissions to other users. If those other users have granted permissions to other users, you must use the cascade option to revoke permissions from them as well. The user specified in *name_list* retains permission to access the object, but can no longer grant access to other users. grant option for applies only to object access permissions, not to object creation permissions.
- The cascade option in a revoke statement removes the specified object access permissions from the user(s) specified in *name_list*, and also from any users they granted those permissions to.

You may only grant and revoke permissions on objects in the current database.

If several users grant access to an object to a particular user, the user's access remains until access is revoked by all those who granted access or until a System Administrator revokes the access. That is, if a System Administrator revokes access, the user is denied access even though other users have granted access.

Only a System Security Officer can grant or revoke permissions to create triggers. The Database Owner can create triggers on any user table. Users can only create triggers on tables that they own.

Permission to issue the create trigger command is granted to users by default.

When the System Security Officer revokes permission for a user to create triggers, a revoke row is added in the sysprotects table for that user. To grant permission to that user to issue create trigger, issue two grant commands: the first command removes the revoke row from sysprotects; the second inserts a grant row. If permission to create triggers is revoked, the user cannot create triggers even on tables that the user owns. Revoking permission to create triggers from a user affects only the database where the revoke command was issued.

Concrete identification

Adaptive Server identifies users during a session by login name. This identification applies to all databases in the server. When the user creates an object, the server associates both the owner's database user ID (*uid*) and the creator's login name with the object in the sysobjects table. This information concretely identifies the object as belonging to that user, which allows the server to recognize when permissions on the object can be granted implicitly.

If an Adaptive Server user creates a table and then creates a procedure that accesses the table, any user who is granted permission to execute the procedure does not need permission to access the object directly. For example, by giving user "mary" permission on proc1, she can see the id and descr columns from table1, though she does not have explicit select permission on the table:

```
create table table1 (id     int,
                     amount money,
                     descr    varchar(100))

create procedure proc1 as select id, descr from
table1

grant execute on proc1 to mary
```

There are, however, some cases where implicit permissions are only useful if the objects can be concretely identified. One case is where aliases and cross-database object access are both involved.

You cannot drop an alias if the aliased login created any objects or thresholds. Before using sp_dropalias to remove an alias that has performed these actions, remove the objects or procedures. If you still need them after dropping the alias, recreate them with a different owner.

Special requirements for SQL92 standard compliance

When you have used the set command to turn ansi_permissions on, additional permissions are required for update and delete statements. Table 11-3 summarizes the required permissions.

Table 11-3: ANSI permissions for update and delete

	Permissions required: *set ansi_permissions off*	**Permissions required: *set ansi_permissions on***
update	update permission on columns where values are being set	update permission on columns where values are being set and select permission on all columns appearing in the where clause select permission on all columns on the right side of the set clause
delete	delete permission on the table	delete permission on the table from which rows are being deleted and select permission on all columns appearing in the where clause

If ansi_permissions is on and you attempt to update or delete without having all the additional select permissions, the transaction is rolled back and you receive an error message. If this occurs, the object owner must grant you select permission on all relevant columns.

Examples of granting object access permissions

This statement gives Mary and the "sales" group permission to insert into and delete from the titles table:

```
grant insert, delete
on titles
to mary, sales
```

This statement gives Harold permission to use the stored procedure makelist:

```
grant execute
on makelist
to harold
```

This statement grants permission to execute the custom stored procedure sa_only_proc to users who have been granted the System Administrator role:

```
grant execute
on sa_only_proc
to sa_role
```

This statement gives Aubrey permission to select, update, and delete from the authors table and to grant the same permissions to other users:

```
grant select, update, delete
on authors
to aubrey
with grant option
```

Examples of revoking object access permissions

These two statements both revoke permission for all users except the table owner to update the price and total_sales columns of the titles table:

```
revoke update
on titles (price, total_sales)
from public
revoke update(price, total_sales)
on titles
from public
```

This statement revokes permission from Clare to update the authors table and simultaneously revokes that permission from all users to whom she had granted that permission:

```
revoke update
on authors
from clare
cascade
```

This statement revokes permission from operators to execute the custom stored procedure new_sproc:

```
revoke execute
on new_sproc
from oper_role
```

Granting and revoking object creation permissions

Object creation permissions regulate the use of commands that create objects. These permissions can be granted only by a System Administrator or a Database Owner.

The object creation commands are:

create database
create default
create procedure
create rule
create table
create view

The syntax for object creation permissions differs slightly from the syntax for object access permissions. The syntax for grant is:

```
grant {all [privileges] | command_list}
    to {public | name_list | role_name}
```

The syntax for revoke is:

```
revoke {all [privileges] | command_list}
    from {public | name_list | role_name}
```

where:

- all or all privileges can be used only by a System Administrator or the Database Owner. When used by a System Administrator in the master database, grant all assigns all create permissions, including create database. If the System Administrator executes grant all from another database, all create permissions are granted except create database. When the Database Owner uses grant all, Adaptive Server grants all create permissions except create database, and prints an informational message.
- *command_list* is the object creation permissions that you are granting or revoking. Separate commands with commas. The list can include create database, create default, create procedure, create rule, create table, and create view. create database permission can be granted only by a System Administrator, and only from within the master database.
- public is all users except the Database Owner (who "owns" object creation permissions within the database).
- name_list is a list of user or group names, separated by commas.
- *role_name* is the name of an Adaptive Server system or user-defined role. You can create and define a hierarchy of user-defined roles and grant them privileges based on the specific role granted.

Examples of granting object creation permissions

The first example grants Mary and John permission to use create database and create table. Because create database permission is being granted, this command can only be executed by a System Administrator within the master database. Mary and John's create table permission applies only to the master database.

```
grant create table, create database
to mary, john
```

This command grants permission to create tables and views in the current database to all users:

```
grant create table, create view
to public
```

Example of revoking object creation permissions

This example revokes permission to create tables and rules from "mary:"

```
revoke create table, create rule
from mary
```

Combining *grant* and *revoke* statements

You can assign specific permissions to specific users, or, if most users are going to be granted most privileges, it may be easier to assign all permissions to all users, and then revoke specific permissions from specific users.

For example, a Database Owner can grant all permissions on the titles table to all users by issuing:

```
grant all
on titles
to public
```

The Database Owner can then issue a series of revoke statements, for example:

```
revoke update
on titles (price, advance)
from public
revoke delete
on titles
from mary, sales, john
```

grant and revoke statements are order-sensitive: in case of a conflict, the most recently issued statement supersedes all others.

Note Under SQL rules, you must use the grant command before using the revoke command, but the two commands cannot be used within the same transaction. Therefore, when you grant "public" access to objects, and then revoke that access from an individual, there is a short period of time during which the individual has access to the objects in question. To prevent this situation, use the create schema command to include the grant and revoke clauses within one transaction.

Understanding permission order and hierarchy

grant and revoke statements are sensitive to the order in which they are issued. For example, if Jose's group has been granted select permission on the titles table and then Jose's permission to select the advance column has been revoked, Jose can select all the columns except advance, while the other users in his group can still select all the columns.

A grant or revoke statement that applies to a group or role changes any conflicting permissions that have been assigned to any member of that group or role. For example, if the owner of the titles table has granted different permissions to various members of the sales group, and wants to standardize, he or she might issue the following statements:

```
revoke all on titles from sales
grant select on titles(title, title_id, type,
        pub_id)
    to sales
```

Similarly, a grant or revoke statement issued to public will change, for all users, all previously issued permissions that conflict with the new regime.

The same grant and revoke statements issued in different orders can create entirely different situations. For example, the following set of statements leaves Jose, who belongs to the public group, without any select permission on titles:

```
grant select on titles(title_id, title) to jose
revoke select on titles from public
```

In contrast, the same statements issued in the opposite order result in only Jose having select permission and only on the title_id and title columns:

```
revoke select on titles from public
grant select on titles(title_id, title) to jose
```

When you use the keyword public with grant, you are including yourself. With revoke on object creation permissions, you are included in public unless you are the Database Owner. With revoke on object access permissions, you are included in public unless you are the object owner. You may want to deny yourself permission to use your own table, while giving yourself permission to access a view built on it. To do this, you must issue grant and revoke statements explicitly setting your permissions. You can reinstitute the permission with a grant statement.

Granting and revoking roles

After a role is defined, it can be granted to any login account or role in the server, provided that it does not violate the rules of mutual exclusivity and hierarchy. Table 11-4 lists the tasks related to roles, the role required to perform the task, and the command to use.

Table 11-4: Tasks, required roles, and commands to use

Task	Required Role	Command
Grant the sa_role role	System Administrator	grant role
Grant the sso_role role	System Security Officer	grant role
Grant the oper_role role	System Security Officer	grant role
Grant user-defined roles	System Security Officer	grant role
Create role hierarchies	System Security Officer	grant role
Modify role hierarchies	System Security Officer	revoke role
Revoke system roles	System Security Officer	revoke role
Revoke user-defined roles	System Security Officer	revoke role

Granting roles

To grant roles to users or other roles, use:

grant role role_granted [{, *role_granted*}...]
to *grantee* [{, *grantee*}...]

where:

- *role_granted* is the role being granted. You can specify any number of roles to be granted.
- *grantee* is the name of the user or role. You can specify any number of grantees.

All roles listed in the grant statement are granted to all grantees. If you grant one role to another, it creates a role hierarchy.

For example, to grant Susan, Mary, and John the "financial_analyst" and the "payroll_specialist" roles, enter:

```
grant role financial_analyst, payroll_specialist
  to susan, mary, john
```

Understanding *grant* and roles

You can use the grant command to grant permission on objects to all users who have been granted a specified role, whether system or user-defined. This allows you to restrict use of an object to users who have been granted any of these roles:

- System Administrator
- System Security Officer
- Operator
- Any user-defined role

You can also use the grant command to grant a role to a user, another role or roles, or a group.

However, grant permission does not prevent users who do *not* have the specified role from being granted execute permission on a stored procedure. If you want to ensure, for example, that only System Administrators can successfully execute a stored procedure, use the proc_role system function within the stored procedure itself. See "Displaying information about roles" on page 379 for more information.

Permissions granted to roles override permissions granted to users or groups. For example, assume John has been granted the System Security Officer role, and sso_role has been granted permission on the sales table. If John's individual permission on sales is revoked, he is still able to access sales when he has sso_role active because his role permissions override his individual permissions.

In granting permissions, a System Administrator is treated as the object owner. If a System Administrator grants permission on another user's object, the owner's name appears as the grantor in sysprotects and in sp_helprotect output.

If several users grant access to an object to a particular user, the user's access remains until access is revoked by all those who granted access. If a System Administrator revokes access, the user is denied access, even though other users have granted access.

Revoking roles

Use revoke role to revoke roles from users and other roles:

revoke role *role_name* [{, *role_name*}...]from *grantee* [{, *grantee*}...]

where:

- *role_name* is the role being revoked. You can specify any number of roles to be revoked.
- *grantee* is the name of the user or role. You can specify any number of grantees.

All roles listed in the revoke statement are revoked from all grantees.

You cannot revoke a role from a user while the user is logged in.

Row-level access control

Database Owners and table owners can restrict access to a table's data rows by defining access rules and binding those rules to the table. Access to data can be further controlled by setting application contexts and creating login triggers.

These features can be grouped under the concept of row-level access control. Row-level access control enables the Database Owner or table owner to control the rows in a table that users can access, based on their identification or profile and the privileges the user has from the application level. Adaptive Server enforces row-level access control for all data manipulation languages (DMLs), which prevents users from bypassing the access control to get to the data.

Access rules

Domain rules allow table owners to control the values that users can enter into a particular column that is using a base datatype, or any column that is using a user-defined datatype. Rules are enforced during inserts and updates

Adaptive Server enables row-level protection through access rules. Access rules are enforced on select, update, and delete operations. Adaptive Server enforces the access rules on all columns that are read in a query, even if the columns are not included in the select list. In other words, for a given query, Adaptive Server enforces the domain rule on the table that is updated and the access rule on the tables that are read.

Using access rules is similar to using views or an ad hoc query with where clauses, and does not cause performance degradation. The query is compiled and optimized after the access rules are attached. Therefore, if there are indexes on the columns that have access rules, the queries may perform better.

Access rules using Java function and application contexts

Application developers can write flexible access rules using Java and application contexts, described in "Application contexts" on page 417. For example, you can write a rule that is hierarchical. If table T contains all the employees' schedules, then the President can see all employees' schedules. Individual VPs can see their own schedules and their direct reports' work schedules, but not the President's schedule.

Access rules can be bound to user-defined datatypes that are defined using sp_addtype. Adaptive Server enforces the access rule on user tables that use these user-defined datatypes. This relieves the Database Owner and table owner from the task of binding access rules to columns in their normalized schema. For example, there can be a user-defined datatype named username for which the base type is varchar(30). The Database Owner or table owner can create an access rule and bind it to username. The owners can then use username in any tables that their application will use. Adaptive Server enforces the access rule on the tables that have columns of the username datatype.

Syntax for access rules

The access parameter is used in the create rule syntax to allow creation of access rules. For example, a table owner creates and populates table T (username char(30), title char(20), classified_data char(1024)):

```
AA, "Administrative Assistant","Memo to President"
AA, "Administrative Assistant","Tracking Stock Movements"
VP1, "Vice President", "Meeting Schedule"
VP2, "Vice President", "Meeting Schedule"
```

The table owner creates a default and a domain rule on the username column. The domain rule ensures that the column is updated with correct values. If the default and domain rule are not created, there is a potential security problem in which the user can insert a row into the table with arbitrary data that will not be qualified by the access rule.

The table owner then creates an access rule and binds it to the username column using sp_bindrule.

```
create default uname_default
as suser_name()
go

sp_bindefault uname_default, "T.username"
go

*/
create accessrule uname_acc_rule
as @username = suser_name()
go

sp_bindrule uname_acc_rule, "T.username"
go
```

A user issues the following query:

```
select * from T
```

Adaptive Server processes the access rule that is bound to the username column on T and attaches it to the query tree. The tree is then optimized and an execution plan is generated and executed as if the user had executed the query with the filter clause given in the access rule. In other words, Adaptive Server attaches the access rule and executes the query as:

```
select * from T where T.username = suser_name().
```

The result of an Administrative Assistant executing the select query is:

```
AA, "Administrative Assistant","Memo to President"
AA, "Administrative Assistant","Tracking Stock Movements"
```

The where T.username = suser_name() part of the query is enforced by the server. The user cannot bypass the access rule.

Extended access rule syntax

Two kinds of access rules can be created: the AND access rule and the OR access rule. The access rule can be bound to a column, username or a user-defined datatype. Multiple access rules can be bound to the different columns of a table or different datatypes of a table. When the table is accessed, several access rules go into effect and those access rules interact with each other. OR access rules are used together and if any one of the OR access rules on the table are satisfied, the row can be accessed. AND access rule are used together and only if all of the AND access rules are satisfied can the row can be accessed.

Note If there is only one access rule on a row of the table and it is an OR access rule, it behaves as an AND access rule.

Syntax

To create an access rule:

```
create rule [ [ and | or ] access]
    [owner.]rule_name
    as condition_expression
```

To bind datatypes:

```
sp_bindrule rule_name, datatype   -
```

To bind to columns:

```
sp_bindrule rule_name, column
```

To unbind only accessrule if exists using:

```
sp_unbindrule datatype, null, "accessrule"
```

or:

```
sp_unbindrule column, null, "accessrule"
```

To unbind both accessrule and domain_rule if exists using:

```
sp_unbindrule datatype, null, "all"
```

or

```
sp_unbindrule column, null, "all"
```

To unbindrule only accessrule for future only if exists using:

```
sp_unbindrule datatype, futureonly, "accessrule"
```

To unbindrule both accessrule and domain rule for future only if exists using:

sp_unbindrule *datatype*, futureonly, "all"

drop rule *rule_name*

Example

```
create  access rule empid1_access
as @empid = 1

create access rule deptno1_access
as @deptid =2

/*
**      create OR ACCESS rule by
**      "create or access rule rule_name as ..."
*/
create or access rule name1_access
as @name = "smith"

create or access rule phone_access
as @phone ="9999"

create table testtab1 (empno int, deptno int,name char(10), phone char(4))

/* Bound access rule to the columns */
sp_bindrule empid1_access, "testtab1.empno"
Rule bound to table column.
(return status = 0)

sp_bindrule deptno1_access,"testtab1.deptno"
Rule bound to table column.
(return status = 0)

sp_bindrule name1_access,"testtab1.name"
Rule bound to table column.
(return status = 0)

sp_bindrule phone_access,"testtab1.phone"
Rule bound to table column.
(return status = 0)

insert testtab1 values (1,1,"smith","3245")
(1 row affected)

insert testtab1 values(2,1,"jones","0283")
(1 row affected)
```

```
insert testtab1 values(1,2,"smith","8282")
(1 row affected)

insert testtab1 values(2,2,"smith","9999")
(1 row affected)

insert testtab1 values(3,2,"smith","8888")
(1 row affected)

insert testtab1 values(1,2,"jones","9999")
(1 row affected)

insert testtab1 values(2,3,"jones","9999")
(1 row affected)

/** return rows when empno = 1 and deptno = 2
*   and ( name = "smith" or phone = "9999" )
*/
select * from testtab1
 empno       deptno      name       phone
 ----------- ----------- ---------- -----
 1           2           smith      8282
 1           2           jones      9999

(2 rows affected)

/* unbound accessrule from specific column */
sp_unbindrule "testtab1.empno",NULL,"accessrule"
Rule unbound from table column.
(return status = 0)

/** return rows when deptno = 2 and ( name = "smith"
*   or phone = "9999" )
*/
select * from testtab1
 empno       deptno      name       phone
 ----------- ----------- ---------- -----
 1           2           smith      8282
 2           2           smith      9999
 3           2           smith      8888
 1           2           jones      9999

(4 rows affected)

/* unbound all rules from specific column */
```

```
sp_unbindrule "testtab1.deptno",NULL,"all"
Rule unbound from table column.
(return status = 0)

/* return the rows when name = "smith" or phone = "9999" */
select * from testtab1
 empno       deptno      name       phone
 ----------- ----------- ---------- -----
 1           1           smith      3245
 1           2           smith      8282
 2           2           smith      9999
 3           2           smith      8888
 1           2           jones      9999
 2           3           jones      9999

(6 rows affected)
```

Access rules and *alter table* command

When the table owner uses the alter table command, Adaptive Server disables access rules during the execution of the command and enables them upon completion of the command. The access rules are disabled to avoid filtering of the table data during the alter table command.

Access rules and *bcp*

Adaptive Server enforces access rules when data is copied out of a table using the bulk copy utility (bcp). Adaptive Server cannot disable access rules as with alter table, because bcp can be used by any user who has select permission on the table.

For security purposes, the Database Owner should lock the table exclusively and disable access rules during bulk copy out. The lock disables access to other users while the access rules are disabled. The Database Owner should enable the access rules and unlock the table after the data has been copied.

Note If access rules are enabled, bcp out only retrieves as much data as the user who is running bcp has permissions on. If the entire table is to be copied, you must lock the table, drop the access rules, bcp out the data, and then re-apply the access rules and unlock the table.

Access rules example scenarios

For this example, assume there is one domain rule for the region column and an access rule for the custid column, which is not used in this query. For updates, customer_table is read, and then updated. Adaptive Server enforces the access rule while reading customer_table on custid column, and, after updating, enforces the domain rule on the region column.

```
update customer_table
    set region = 'northwest'
    where region = 'north'
```

In this next example, assume there are domain rules on orders_table and access rules on old_orders_table. Adaptive Server enforces the domain rule on orders_table because orders_table is updated, and the access rule on the old_orders_table because old_orders_table is read.

```
insert into orders_table
select *
    from old_orders_table
```

Access rules using Java user-defined functions

Access rules can use user-defined Java functions that use JDBC to look up data in additional tables. Using Java functions, you can, for example, write sophisticated rules that use the profile of the application, the user logged in to use the application, and the roles that the user currently has for the application.

The following Java class uses the GetSecVal method to demonstrate how you can use Java methods as user-defined functions inside access rules.

```
import java.sql.*;
import java.util.*;

public class sec_class {
static String _url = "jdbc:sybase:asejdbc";
public static int GetSecVal(int c1)
{
try
{
PreparedStatement pstmt;
ResultSet rs = null;
Connection con = null;
int pno_val;

pstmt = null;
```

```
Class.forName("sybase.asejdbc.ASEDriver");
con = DriverManager.getConnection(_url);

if (con == null)
{
return (-1);
}

pstmt = con.prepareStatement("select classification from sec_tab where id = ?");

if (pstmt == null)
{
return (-1);
}

pstmt.setInt(1, c1);

rs = pstmt.executeQuery();

rs.next();

pno_val = rs.getInt(1);

rs.close();

pstmt.close();

con.close();

return (pno_val);

}
catch (SQLException sqe)
{
return(sqe.getErrorCode());
}
catch (ClassNotFoundException e)
{

System.out.println("Unexpected exception : " + e.toString());
System.out.println("\nThis error usually indicates that " + "your Java CLASSPATH
environment has not been set properly.");
e.printStackTrace();
return (-1);
}
```

```
catch (Exception e)
{
System.out.println("Unexpected exception : " + e.toString());
e.printStackTrace();
return (-1);
}
}
}

(from Shell)
javac sec_class.java

jar cufo sec_class. jar sec_class.class
installjava -Usa -Password -f/work/work/FGAC/sec_class.jar -
-D testdb

(from isql)

create table sec_tab (id int, classification int)
go
insert into sec_tab values (1,10)
insert into sec_tab values (2,9)
insert into sec_tab values (3,7)
insert into sec_tab values (4,7)
insert into sec_tab values (5,4)
insert into sec_tab values (6,4)
insert into sec_tab values (7,4)
go

sp_addtype class_level, int
go

create table sec_data (c1 varchar(30),
c2 varchar(30),
c3 varchar(30),
clevel class_level)
go

declare @v1 int
select @v1 = 5
while @v1 > 0
begin
insert into sec_data values('8', 'aaaaaaaaaa', 'aaaaaaaaaa', 8)
insert into sec_data values('7', 'aaaaaaaaaa', 'aaaaaaaaaa', 7)
insert into sec_data values('5', 'aaaaaaaaaa', 'aaaaaaaaaa', 5)
insert into sec_data values('5', 'aaaaaaaaaa', 'aaaaaaaaaa', 5)
insert into sec_data values('2', 'aaaaaaaaaa', 'aaaaaaaaaa', 2)
insert into sec_data values('3', 'aaaaaaaaaa', 'aaaaaaaaaa', 3)
```

```
select @v1 = @v1 -1
end
go

create access rule clevel as
@clevel <= sec_class.GetSecVal(suser_id())
go

create default clevel_def as sec_class.GetSecVal(suser_id())
go

sp_bindefault clevel_def, class_level
go

sp_bindrule clevel, class_level
go

grant all on sec_data to public
go
grant all on sec_tab to public
go
```

Application contexts

Applications on a database server should be programmed to limit access to the data based on their users and user profiles.

The application developer is responsible for coding the application appropriately. For example, a Human Resources application is programmed to know which users are allowed to update salary information.

Application contexts allow users to define, store, and retrieve user profiles–the roles each user is authorized to use and groups to which he or she belongs–and the application currently in use by each user. Application contexts can be used to store and retrieve arbitrary client data, and can use Adaptive Server to store client information.

Application contexts are specific to a session. They are not persistent across sessions; however, they are available across nested levels of statement execution, unlike local variables.

An application context consists of a *context name*, an *attribute name* , and an *attribute value*. Users define the *context name*, the *attributes* and values for each context. Sybase provides a variety of attributes in the system application context, sys_session. For details, see Number 11 under Examples.

You can also create your own application contexts as described in "Creating and maintaining application contexts" on page 419.

Setting permissions for using application context functions

Application contexts are set, retrieved, and removed using functions. This means any user who is logged in can reset the profiles of the session. Although execution of a function is audited, security may be compromised before the problem is noticed. You can restrict access to functions using grant and revoke privileges. Only the application context functions perform data access control checks on the user.

Granting or revoking privileges for other functions does not have any effect in Adaptive Server.

Application context function execution is treated as a select data manipulation language. The owner of the function is the System Administrator of the server. Only users with the sa_role can grant or revoke privileges on the functions. Only the select privilege is checked as part of server-enforced data access control checks done by the functions.

By default, privileges on the functions are revoked to PUBLIC. This matches current defaults for table-level privileges.

You can grant and revoke privileges to users, roles, and groups in a given database for objects in that database. The only exceptions are create database, set session authorization, and connect. A user granted these privileges should be a valid user in the master database. For other privileges, the user should be valid in the database where the object is located.

However, functions do not have an object ID nor they do not have a home database. This means that in each database, the Database Owner must grant to the appropriate user the select privilege for the functions. Adaptive Server finds the user's default database and checks the permissions against this database. With this approach, only the owner of the users' default database needs to grant the select privilege. If other databases need to be restricted, the owner of those databases must explicitly revoke permission for the user in those databases.

A System Administrator can execute the following commands to grant or revoke select privileges on specific application context functions:

set_appcontext where *context_name* and *attribute_name* have datatypes of char(30).

If the attribute does not exist in the application context, get_appcontext returns NULL.

The attribute value is returned as a char datatype.

If the rule must use the attribute value to compare against other datatypes, then the rule should convert the char data to the appropriate datatype, list_appcontext rm_appcontext.

Creating and maintaining application contexts

The following functions are available for creating and maintaining application contexts:

```
grant select on set_appcontext to user_role
grant select on set_appcontext to joe_user
revoke select on set_appcontext from joe_user
set_appcontext
set_appcontext is used to set a context name, attribute name, and attribute
value for the user session.
set_appcontext (“context_name”, “attribute_name”, “attribute_value”)
```

Where *context_name* and *attribute_name* have datatypes of char(30) and *attribute_value* has a datatype of char(2048). This function returns 0 for success and -1 for failure. set_appcontext cannot override values of an existing application context. If you want to assign new values to a context, remove the context, then re-create it with the new values. If the values being set already exist in the session, the function returns -1. Attributes are saved as char datatype. If the rule must use the attribute value to compare against other datatypes, the rule should convert the char data to the appropriate datatype

Examples

This example shows set_appcontext with a datatype conversion included in the value:

```
select set_appcontext("CONTEXT1","ATTR1", "VALUE1")
------------
0
This example creates an application context named CONTEXT1. The attribute
is named ATTR1 with a value of VALUE1.
select set_appcontext("CONTEXT1", "ATTR2", convert(char(20), @numericvar)
```

```
-----------
0
```

This example shows the result of attempting to override an existing application context. This context was created in example 1. The context must be removed, then re-created with the new values:

```
select set_appcontext("CONTEXT1", "ATTR1", "VALUE1")
----------
-1
```

This example shows the result of a user without appropriate permissions attempting to set the application context:

```
select set_appcontext("CONTEXT1", "ATTR2", "VALUE1")
Select permission denied on function set_appcontext, database dbid
set_appcontext
get_appcontext returns the value of the attribute in a given context.
get_appcontext (“context_name”, “attribute_name”)
```

Where *context_name* and *attribute_name* have datatypes of char(30). If the attribute does not exist in the application context, get_appcontext returns “null.” The attribute value is returned as a char datatype. If the rule must use the attribute value to compare against other datatypes, then the rule should convert the char data to the appropriate datatype.

Acquiring the permissions of another user

Adaptive Server provides two ways of acquiring another user’s identity and permissions status:

- A Database Owner can use the setuser command to “impersonate” another user’s identity and permissions status in the current database. See “Using setuser” on page 420.
- **proxy authorization** allows one user to assume the identity of another user on a server-wide basis. See “Using proxy authorization” on page 422.

Using setuser

A Database Owner may use setuser to:

- Access an object owned by another user
- Grant permissions on an object owned by another user
- Create an object that will be owned by another user
- Temporarily assume the DAC permissions of another user for some other reason

While the setuser command enables the Database Owner to automatically acquire another user's DAC permissions, the command does not affect the roles that have been granted.

setuser permission defaults to the Database Owner and cannot be transferred. The user being impersonated must be an authorized user of the database. Adaptive Server checks the permissions of the user being impersonated.

System Administrators can use setuser to create objects that will be owned by another user. However, System Administrators operate outside the DAC permissions system; therefore, they need not use setuser to acquire another user's permissions. The setuser command remains in effect until another setuser command is given, the current database is changed, or the user logs off.

The syntax is:

setuser ["*user_name*"]

where *user_name* is a valid user in the database that is to be impersonated.

To reestablish your original identity, use setuser with no value for *user_name*.

This example shows how the Database Owner would grant Joe permission to read the authors table, which is owned by Mary:

```
setuser "mary"

grant select on authors to joe

setuser     /*re-establishes original identity*/
```

Using proxy authorization

With the proxy authorization capability of Adaptive Server, System Security Officers can grant selected logins the ability to assume the security context of another user, and an application can perform tasks in a controlled manner on behalf of different users. If a login has permission to use proxy authorization, the login can impersonate any other login in Adaptive Server.

Warning! The ability to assume another user's identity is extremely powerful and should be limited to trusted administrators and applications. A user with this permission can even assume the identity of the "sa" login, and, thereby, have unlimited power within Adaptive Server.

A user executing set proxy or set session authorization operates with both the login name and server user ID of the user being impersonated. The login name is stored in the name column of master..syslogins and the server user ID is stored in the suid column of master..syslogins. These values are active across the entire server in all databases.

Note set proxy and set session authorization are identical in function and can be used interchangeably. The only difference between them is that set session authorization is ANSI SQL92 compatible, and set proxy is a Transact-SQL extension.

Granting proxy authorization

System Security Officers use the grant set proxy or grant set session authorization command to give a user permission to impersonate another user within the server. The user with this permission can then execute either set proxy or set session authorization to become another user.

To grant proxy authorization permission, you must be a System Security Officer and execute the grant command from the master database. The syntax is:

```
grant set proxy
    to {public | name_list | role_name}
```

or

```
grant set session authorization
    to {public | name_list | role_name}
```

where:

- *public* is all users. Sybase recommends that you not grant this permission to "public."
- *role_name* is an Adaptive Server system or user-defined role. You can grant permissions to users based on the specific role granted.
- *name_list* is user database or group names, separated by commas. The user must be a valid user in the master database.

To grant set proxy to an application with the login "appl" if you do not have sso_role currently active, and you are not in the master database, execute:

```
use master
go
set role sso_role on
go
grant set proxy to appl
go
```

To grant set proxy to that user-defined role "accountant," execute:

```
grant set proxy to accountant
```

To grant set session authorization to the "sa" account, whose user name in every database is "dbo," execute:

```
grant set proxy to dbo
```

Executing proxy authorization

Follow these rules when you execute set proxy or set session authorization:

- You cannot execute set proxy or set session authorization from within a transaction.
- You cannot use a locked login for the proxy of another user. For example, if "joseph" is a locked login, the following command is not allowed:

  ```
  set proxy "joseph"
  ```

- You can execute set proxy or set session authorization from any database that you are allowed to use. However, the *login_name* you specify must be a valid user in the database, or the database must have a "guest" user defined for it.
- Only one level is permitted; to impersonate more than one user, you must return to your original identity between impersonations.

- If you execute set proxy or set session authorization from within a procedure, your original identity is automatically resumed when you exit the procedure.

If you have a login that has been granted permission to use set proxy or set session authorization, you can set proxy to impersonate another user. The syntax is:

set proxy *login_name*

or

set session authorization *login_name*

where *login_name* is the name of a valid login in master..syslogins. Enclose the login name in quotation marks.

For example, to set proxy to "mary," execute:

```
set proxy mary
```

After setting proxy, check your login name in the server and your user name in the database. For example, assume that your login is "ralph" and that you have been granted set proxy authorization. You want to execute some commands as "sallyn" and as "rudolph" in pubs2 database. "sallyn" has a valid name ("sally") in the database, but Ralph and Rudolph do not. However, pubs2 has a guest user defined. You can execute:

```
set proxy "sallyn"
go
use pubs2
go
select suser_name(), user_name()
go
------------------------------ ------------------
sallyn                         sally
```

To change to Rudolph, you must first change back to your own identity. To do so, execute:

```
set proxy "ralph"
select suser_name(), user_name()
go
------------------------------ ------------------
ralph                          guest
```

Notice that Ralph is a "guest" in the database.

Then execute:

```
set proxy "rudolph"
```

```
go
select suser_name(), user_name()
go
------------------------------ --------------------
rudolph                        guest
```

Rudolph is also a guest in the database because Rudolph is not a valid user in the database.

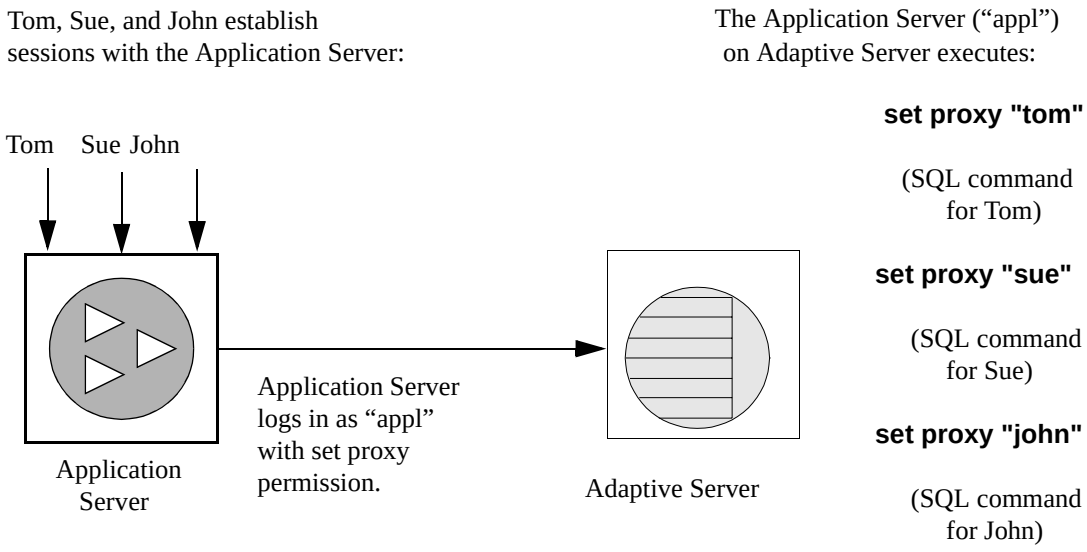
Now, impersonate the "sa" account. Execute:

```
set proxy "ralph"
go
set proxy "sa"
go
select suser_name(), user_name()
go
-------------------------- --------------------
sa                         dbo
```

Proxy authorization for applications

Figure 11-1 shows an application server logging in to Adaptive Server with the generic login "appl" to execute procedures and commands for several users. While "appl" impersonates Tom, the application has Tom's permissions. Likewise, when "appl" impersonates Sue and John, the application has only Sue's and John's permissions, respectively.

Figure 11-1: Applications and proxy authorization



Reporting on permissions

Table 11-5 lists the system procedures for reporting information about proxies, object creation and object access permissions:

Table 11-5: System procedures for reporting on permissions

To Report Information On	Use
Proxies	system tables
Users and processes	sp_who
Permissions on database objects or users	sp_helprotect
Permissions on specific tables	sp_table_privileges
Permissions on specific columns in a table	sp_column_privileges

Querying the *sysprotects* table for proxy authorization

To display information about permissions that have been granted to, or revoked from, users, groups, and roles, query the sysprotects table. The action column specifies the permission. For example, the action value for set proxy or set session authorization is equal to 167.

You might execute this query:

```
select * from sysprotects where action = 167
```

The results provide the user ID of the user who granted or revoked the permission (column grantor), the user ID of the user who has the permission (column uid), and the type of protection (column protecttype). The protecttype column can contain these values:

- 0 for grant with grant
- 1 for grant
- 2 for revoke

For more information about the sysprotects table, see the *Adaptive Server Reference Manual*.

Displaying information about users and processes

sp_who displays information about all current Adaptive Server users and processes or about a particular user or process. The results of sp_who include the loginame and origname. If a user is operating under a proxy, origname contains the name of the original login. For example, assume that "ralph" executed:

```
set proxy susie
```

and then executes some SQL commands.

sp_who returns “susie” for loginame and “ralph” for origname.

sp_who queries the master..sysprocesses system table, which contains columns for the server user ID (suid) and the original server user ID (origsuid).

For more information, see sp_who in the *Adaptive Server Reference Manual*.

Reporting permissions on database objects or users

Use sp_helprotect to report on permissions by database object or by user, and (optionally) by user for a specified object. Any user can execute this procedure. The syntax is:

```
sp_helprotect [name [, username [, "grant"
    [,"none"|"granted"|"enabled"|role_name]]]]
```

where:

- *name* is either the name of the table, view, or stored procedure, or the name of a user, group, or role in the current database. If you do not provide a name, sp_helprotect reports on all permissions in the database.
- *username* is a user’s name in the current database.

If you specify *username*, only that user’s permissions on the specified object are reported. If *name* is not an object, sp_helprotect checks whether *name* is a user, group, or role and if it is, lists the permissions for the user, group, or role. If you specify keyword grant, and *name* is not an object, sp_helprotect displays all permissions granted by with grant option.

grant displays the permissions granted to *name* with grant option.

none ignores roles granted to the user.

granted includes information on all roles granted to the user.

enabled includes information on all roles activated by the user.

role_name displays permission information for the specified role only, regardless of whether this role has been granted to the user.

For example, suppose you issue the following series of grant and revoke statements:

```
grant select on titles to judy
grant update on titles to judy
```

```
revoke update on titles(contract) from judy
grant select on publishers to judy
    with grant option
```

To determine the permissions Judy now has on each column in the titles table, enter:

```
sp_helprotect titles, judy
```

```
grantor grantee type   action   object     column      grantable
------- ------ -----   ------   ------     ------      -------
dbo     judy   Grant   Select   titles     All         FALSE
dbo     judy   Grant   Update   titles     advance     FALSE
dbo     judy   Grant   Update   titles     notes       FALSE
dbo     judy   Grant   Update   titles     price       FALSE
dbo     judy   Grant   Update   titles     pub_id      FALSE
dbo     judy   Grant   Update   titles     pubdate     FALSE
dbo     judy   Grant   Update   titles     title       FALSE
dbo     judy   Grant   Update   titles     title_id    FALSE
dbo     judy   Grant   Update   titles     total_sales FALSE
dbo     judy   Grant   Update   titles     type        FALSE
```

The first row shows that the Database Owner ("dbo") gave Judy permission to select all columns of the titles table. The rest of the lines indicate that she can update only the columns listed in the display. Judy cannot give select or update permissions to any other user.

To see Judy's permissions on the publishers table, enter:

```
sp_helprotect publishers, judy
```

In this display, the grantable column indicates TRUE, meaning that Judy can grant the permission to other users.

```
grantor grantee type   action   object     column      grantable
------- ------ -----   ------   ------     ------      -------
dbo     judy   Grant   Select   publishers all         TRUE
```

Reporting permissions on specific tables

Use sp_table_privileges to return permissions information about a specified table. The syntax is:

sp_table_privileges *table_name* [, *table_owner* [, *table_qualifier*]]

where:

- *table_name* is the name of the table. It is required.

- *table_owner* can be used to specify the name of the table owner, if it is not “dbo” or the user executing sp_table_privileges.
- *table_qualifier* is the name of the current database.

Use null for parameters that you want to skip.

For example, the following statement:

```
sp_table_privileges titles
```

returns information about all permissions granted on the titles table. For more information about the output of sp_table_privileges see the *Adaptive Server Reference Manual*.

Reporting permissions on specific columns

Use sp_column_privileges to return information about permissions on columns in a table. The syntax is:

sp_column_privileges *table_name* [, *table_owner* [, *table_qualifier* [, *column_name*]]]

where:

- *table_name* is the name of the table.
- *table_owner* can be used to specify the name of the table owner, if it is not “dbo” or the user executing sp_column_privileges.
- *table_qualifier* is the name of the current database.
- *column_name* is the name of the column on which you want to see permissions information.

Use null for parameters that you want to skip.

For example, the following statement:

```
sp_column_privileges publishers, null, null, pub_id
```

returns information about the pub_id column of the publishers table. For more information about the output of sp_column_privileges, see the *Adaptive Server Reference Manual*.

Using views and stored procedures as security mechanisms

Views and stored procedures can serve as security mechanisms. You can give users controlled access to database objects via a view or stored procedure without granting them direct access to the data. For example, you might give a clerk execute permission on a procedure that updates cost information in a projects table without letting the user see confidential data in the table. To use this feature, you must own the procedure or view as well as its underlying objects. If you do not own the underlying objects, users must have permission to access the objects. For more information about when permissions are required, see "Understanding ownership chains" on page 434.

Adaptive Server makes permission checks, as required, when the view or procedure is used. When you create the view or procedure, Adaptive Server makes no permission checks on the underlying objects.

Using views as security mechanisms

Through a view, users can query and modify only the data they can see. The rest of the database is neither visible nor accessible.

Permission to access the view must be explicitly granted or revoked, regardless of the permissions on the view's underlying tables. If the view and underlying tables are owned by the same owner, no permissions need to be given to the underlying tables. Data in an underlying table that is not included in the view is hidden from users who are authorized to access the view but not the underlying table.

By defining different views and selectively granting permissions on them, a user (or any combination of users) can be restricted to different subsets of data. Access can be restricted to:

- A subset of the rows of a base table (a value-dependent subset). For example, you might define a view that contains only the rows for business and psychology books to keep information about other types of books hidden from some users.
- A subset of the columns of a base table (a value-independent subset). For example, you might define a view that contains all the rows of the titles table, but omits the price and advance columns, since this information is sensitive.

- A row-and-column subset of a base table.
- The rows that qualify for a join of more than one base table. For example, you might define a view that joins the titles, authors, and titleauthor tables. This view would hide personal data about authors and financial information about the books.
- A statistical summary of data in a base table. For example, you might define a view that contains only the average price of each type of book.
- A subset of another view, or of some combination of views and base tables.

Let's say you want to prevent some users from accessing the columns in the titles table that display money and sales amounts. You could create a view of the titles table that omits those columns, and then give all users permission on the view but only the Sales Department permission on the table:

```
grant all on bookview to public
grant all on titles to sales
```

An equivalent way of setting up these privilege conditions, without using a view, is to use the following statements:

```
grant all on titles to public
revoke select, update on titles (price, advance,
    total_sales)
from public
grant select, update on titles (price, advance,
    total_sales)
to sales
```

One possible problem with the second solution is that users not in the sales group who enter the command:

```
select * from titles
```

might be surprised to see the message that includes the phrase:

```
permission denied
```

Adaptive Server expands the asterisk into a list of all the columns in the titles table, and since permission on some of these columns has been revoked from non-sales users, access to these columns is denied. The error message lists the columns for which the user does not have access.

To see all the columns for which they do have permission, the non-sales users would have to name them explicitly. For this reason, creating a view and granting the appropriate permissions on it is a better solution.

You can also use views for **context-sensitive protection**. For example, you can create a view that gives a data entry clerk permission to access only those rows that he or she has added or updated. To do so, add a column to a table in which the user ID of the user entering each row is automatically recorded with a default. You can define this default in the create table statement, like this:

```
create table testtable
    (empid       int,
    startdate    datetime,
    username     varchar(30) default user)
```

Next, define a view that includes all the rows of the table where uid is the current user:

```
create view context_view
as
    select *
    from testtable
    where username = user_name()
with check option
```

The rows retrievable through this view depend on the identity of the person who issues the select command against the view. By adding with check option to the view definition, you make it impossible for any data entry clerk to falsify the information in the username column.

Using stored procedures as security mechanisms

If a stored procedure and all underlying objects are owned by the same user, that owner can grant users permission to use the procedure without granting permissions on the underlying objects. For example, you might give a user permission to execute a stored procedure that updates a row-and-column subset of a specified table, even though that user does not have any other permissions on that table.

Roles and stored procedures

Use the grant execute command to grant execute permission on a stored procedure to all users who have been granted a specified role. revoke execute removes this permission. But grant execute permission does not prevent users who do *not* have the specified role from being granted execute permission on the stored procedure.

For further security, you can restrict the use of a stored procedure by using the proc_role system function within the procedure to guarantee that a procedure can be executed only by users who have a given role. proc_role returns 1 if the user has a specific role (sa_role, sso_role, oper_role, or any user-defined role) and returns 0 if the user does not have that role. For example, here is a procedure that uses proc_role to see if the user has the System Administrator role:

```
create proc test_proc
as
if (proc_role("sa_role") = 0)
begin
    print "You don’t have the right role"
    return -1
end
else
    print "You have SA role"
    return 0
```

See “System Functions” in the Adaptive Server Reference Manual for more information about proc_role.

Understanding ownership chains

Views can depend on other views and/or tables. Procedures can depend on other procedures, views, and/or tables. These dependencies can be thought of as an *ownership chain*.

Typically, the owner of a view also owns its underlying objects (other views and tables), and the owner of a stored procedure owns all the procedures, tables, and views referenced by the procedure.

A view and its underlying objects are usually all in the same database, as are a stored procedure and all the objects it references; however, this is not required. If objects are in different databases, a user wanting to use the view or stored procedure must be a valid user or guest user in all of the databases containing the objects. This prevents users from accessing a database unless the Database Owner has authorized it.

When a user who has been granted execute permission on a procedure or view uses it, Adaptive Server does not check permissions on any of the underlying objects if:

- These objects and the view or procedure are owned by the same user, and
- The user accessing the view or procedure is a valid user or guest user in each of the databases containing the underlying objects.

However, if all objects are not owned by the same user, Adaptive Server checks object permissions when the ownership chain is broken. That is, if object A references object B, and B is not owned by the user who owns object A, Adaptive Server checks the permissions for object B. In this way, Adaptive Server allows the owner of the original data to retain control over who is authorized to access it.

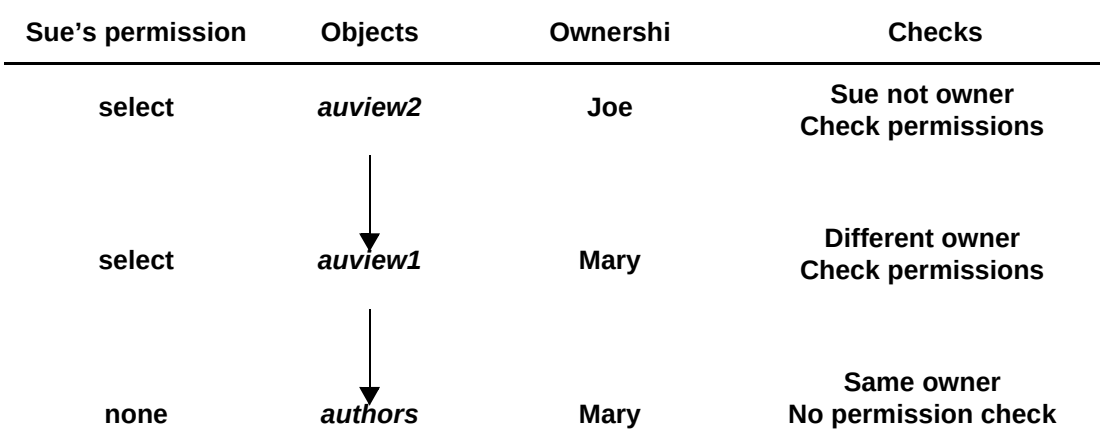
Ordinarily, a user who creates a view needs worry only about granting permissions on that view. For example, say Mary has created a view called auview1 on the authors table, which she also owns. If Mary grants select permission to Sue on auview1, Adaptive Server will let Sue access it without checking permissions on authors.

However, a user who creates a view or stored procedure that depends on an object owned by another user must be aware that any permissions he or she grants depend on the permissions allowed by those other owners.

Example of views and ownership chains

Say Joe creates a view called auview2, which depends on Mary's view auview1. Joe grants Sue select permission on auview2.

Figure 11-2: Ownership chains and permission checking for views, case 1

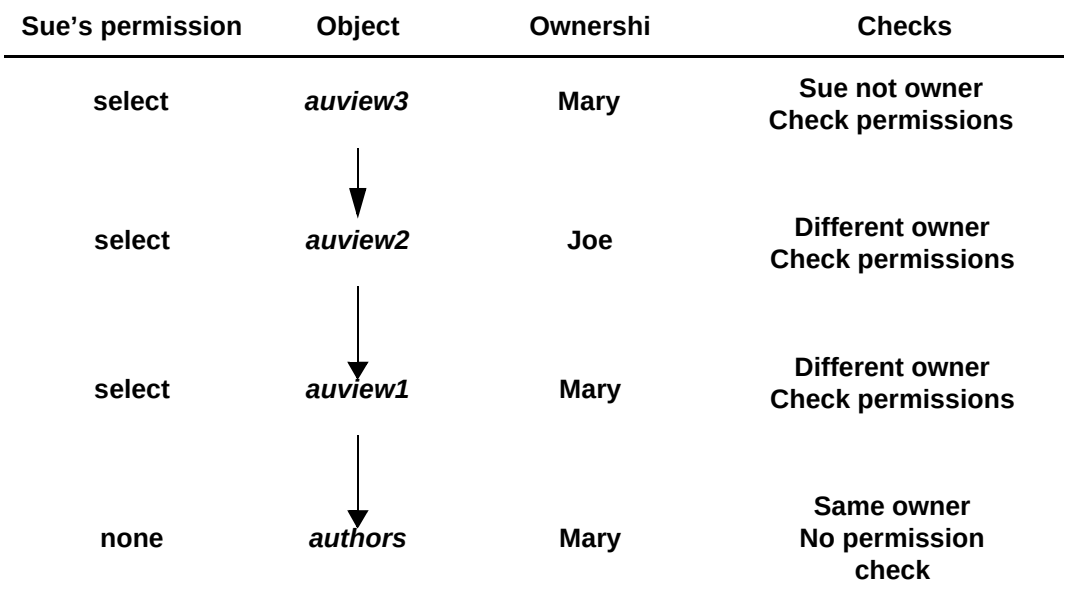


Adaptive Server checks the permissions on auview2 and auview1, and finds that Sue can use them. Adaptive Server checks ownership on auview1 and authors and finds that they have the same owner. Therefore, Sue can use auview2.

Taking this example a step further, suppose that Joe's view, auview2, depends on auview1, which depends on authors. Mary decides she likes Joe's auview2 and creates auview3 on top of it. Both auview1 and authors are owned by Mary.

The ownership chain looks like this:

Figure 11-3: Ownership chains and permission checking for views, case 2

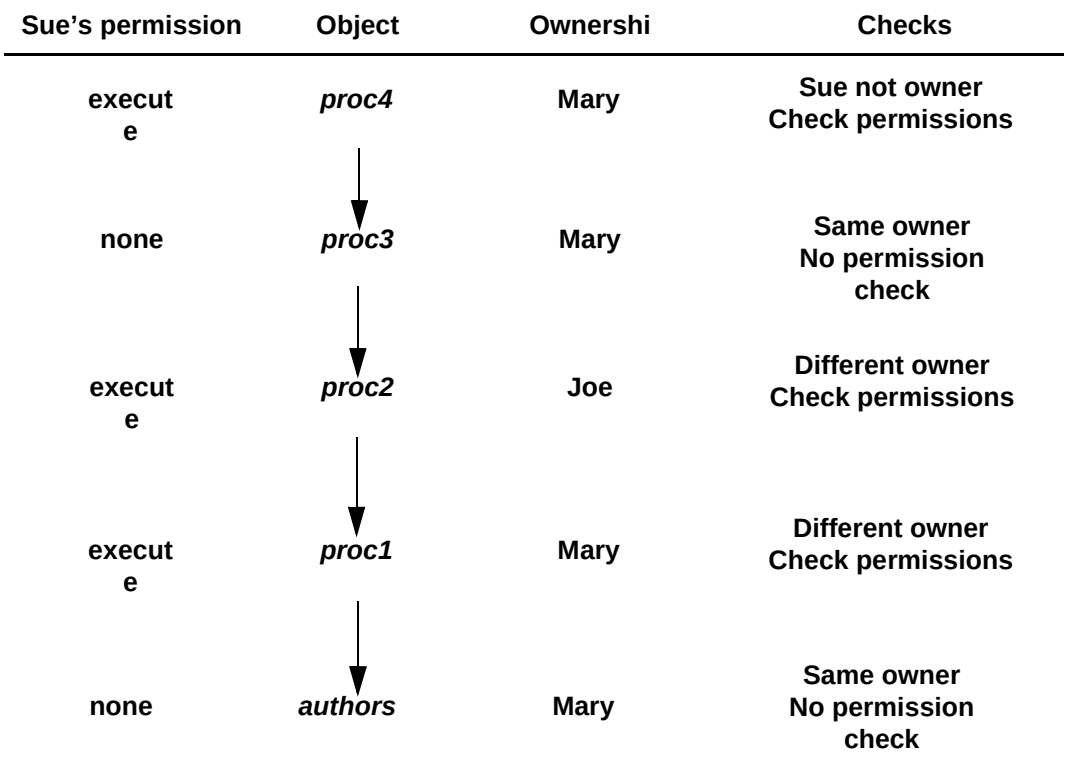


When Sue tries to access auview3, Adaptive Server checks permissions on auview3, auview2, and auview1. If Joe has granted permission to Sue on auview2 and Mary has granted her permission on auview3 and auview1, Adaptive Server allows the access. Adaptive Server checks permissions only if the object immediately before it in the chain has a different owner (or if it is the first object in the chain). For example, it checks auview2 because the object before it—auview3—is owned by a different user. It does not check permission on authors, because the object that immediately depends on it, auview1, is owned by the same user.

Example of procedures and ownership chains

Procedures follow the same rules as views. For example, suppose the ownership chain looks like this:

Figure 11-4: Ownership chains and permission checking for stored procedures



To execute proc4, Sue must have permission to execute proc4, proc2, and proc1. Permission to execute proc3 is not necessary because proc3 and proc4 have the same owner.

Adaptive Server checks Sue's permissions on proc4 and all objects it references each time she executes proc4. Adaptive Server knows which referenced objects to check: it determined this the first time Sue executed proc4, and it saved the information with the procedure's execution plan. Unless one of the objects referenced by the procedure is dropped or redefined, Adaptive Server does not change its initial decision about which objects to check.

This protection hierarchy allows every object's owner to fully control access to the object. Owners can control access to views and stored procedures, as well as to tables.

Permissions on triggers

A **trigger** is a special kind of stored procedure used to enforce integrity, especially referential integrity. Triggers are never executed directly, but only as a side effect of modifying a table. You cannot grant or revoke permissions for triggers.

Only an object owner can create a trigger. However, the ownership chain can be broken if a trigger on a table references objects owned by different users. The protection hierarchy rules that apply to procedures also apply to triggers.

While the objects that a trigger affects are usually owned by the user who owns the trigger, you can write a trigger that modifies an object owned by another user. If this is the case, any users modifying your object in a way that activates the trigger must have permission on the other object as well.

If Adaptive Server denies permission on a data modification command because a trigger affects an object for which the user does not have permission, the entire data modification transaction is rolled back.

For more information on triggers, see the *Transact-SQL User's Guide* or the *Adaptive Server Reference Manual*.

CHAPTER 12 # Auditing

This chapter describes how to set up auditing for your installation.

Topics covered in this chapter include:

Introduction to auditing in Adaptive Server

A principal element of a secure system is accountability. One way to ensure accountability is to audit events on the system. Many events that occur in Adaptive Server can be recorded.

Auditing is an important part of security in a database management system. An audit trail can be used to detect penetration of the system and misuse of resources. By examining the audit trail, a System Security Officer can inspect patterns of access to objects in databases and can monitor the activity of specific users. Audit records are traceable to specific users, which may act as a deterrent to users who are misusing the system.

Each audit record can log the nature of the event, the date and time, the user responsible for it, and the success or failure of the event. Among the events that can be audited are logins and logouts, server boots, use of data access commands, attempts to access particular objects, and a particular user's actions. The **audit trail**, or log of audit records, allows the System Security Officer to reconstruct events that have occurred on the system and evaluate their impact.

The System Security Officer is the only user who can start and stop auditing, set up auditing options, and process the audit data. As a System Security Officer, you can establish auditing for events such as:

- Server-wide, security-relevant events
- Creating, deleting, and modifying database objects
- All actions by a particular user or all actions by users with a particular role active
- Granting or revoking database access
- Importing or exporting data
- Logins and logouts

Correlating Adaptive Server and operating system audit records

The easiest way to link Adaptive Server audit records with operating system records is to make Adaptive Server login names the same as operating system login names.

Alternatively, the System Security Officer can map users' operating system login names to their Adaptive Server login names. However, this approach requires ongoing maintenance, as login names for new users have to be recorded manually.

The audit system

The audit system consists of:

- The sybsecurity database, which contains global auditing options and the audit trail
- The in-memory audit queue, to which audit records are sent before they are written to the audit trail
- Configuration parameters for managing auditing
- System procedures for managing auditing

The *sybsecurity* database

The sybsecurity database is created during the auditing installation process. In addition to all the system tables found in the model database, it contains sysauditoptions, a system table for keeping track of server-wide auditing options, and system tables for the audit trail.

sysauditoptions contains the current setting of global auditing options, such as whether auditing is enabled for disk commands, remote procedure calls, ad hoc user-defined auditing records, or all security-relevant events. These options affect the entire Adaptive Server.

The audit trail

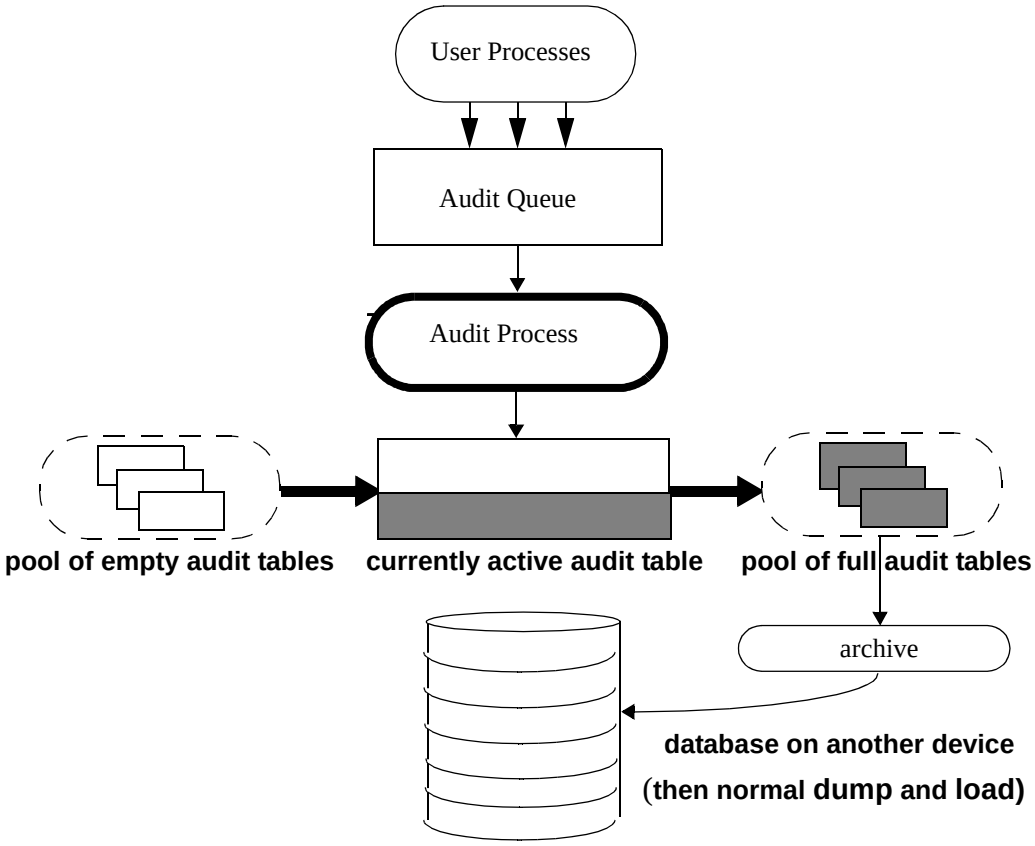
Adaptive Server stores the audit trail in system tables named sysaudits_01 through sysaudits_08. When you install auditing, you determine the number of audit tables for your installation. For example, if you choose to have two audit tables, they are named sysaudits_01 and sysaudits_02. At any given time, only *one* audit table is *current*. Adaptive Server writes all audit data to the current audit table. A System Security Officer can use sp_configure to set, or change, which audit table is current.

The recommended number of tables is two or more with each table on a separate audit device. This allows you to set up a smoothly running auditing process in which audit tables are archived and processed with no loss of audit records and no manual intervention.

Warning! Sybase strongly recommends against using a single audit table on production systems. If you use only a single audit table, you may lose audit records. If you must use only a single audit table, because of limited system resources, refer to "Single-table auditing" on page 458 for instructions.

Figure 12-1 shows how the auditing process works with multiple audit tables.

Figure 12-1: Auditing with multiple audit tables



The auditing system writes audit records from the in-memory audit queue to the current audit table. When the current audit table is nearly full, a threshold procedure can automatically archive the table to another database. The archive database, can be backed up and restored with the dump and load commands. For more information about managing the audit trail, see "Setting up audit trail management" on page 449.

The audit queue

When an audited event occurs, an audit record first goes to the in-memory audit queue. The record remains in memory until the audit process writes it to the audit trail. You can configure the size of the audit queue with the audit queue size parameter of sp_configure.

Before you configure the size of the audit queue, consider the trade-off between the risk of losing records in the queue if the system crashes and the loss of performance when the queue is full. As long as an audit record is in the queue, it can be lost if the system crashes. However, if the queue repeatedly becomes full, overall system performance is affected. If the audit queue is full when a user process tries to generate an audit record, the process sleeps until space in the queue becomes available.

Note Because audit records are not written directly to the audit trail, you cannot count on an audit record's being stored immediately in the current audit table.

Auditing configuration parameters

Use these configuration parameters to manage the auditing process:

- auditing enables or disables auditing for the whole Adaptive Server. The parameter takes effect immediately upon execution of sp_configure. Auditing occurs only when this parameter is enabled.
- audit queue size establishes the size of the audit queue. Because the parameter affects memory allocation, the parameter does not take effect until Adaptive Server is restarted.
- suspend audit when device full controls the behavior of the audit process when an audit device becomes full. The parameter takes effect immediately upon execution of sp_configure.
- current audit table sets the current audit table. The parameter takes effect immediately upon execution of sp_configure.

System procedures for auditing

Use these system procedures to manage the auditing process:

- sp_audit enables and disables auditing options. This is the only system procedure required to establish the events to be audited.
- sp_displayaudit displays the active auditing options.
- sp_addauditrecord adds user-defined audit records (comments) into the audit trail. Users can add these records only if a System Security Officer enables ad hoc auditing with sp_audit.

Installing and setting up auditing

Table 12-1 provides a general procedure for setting up auditing.

Table 12-1: General procedure for auditing

Action	Description	See
1. Install auditing.	Set the number of audit tables and assign devices for the audit trail and the syslogs transaction log in the sybsecurity database.	"Installing the audit system" on page 446 and the Adaptive Server installation and configuration documentation
2. Set up audit trail management.	Write and establish a threshold procedure that receives control when the current audit table is nearly full. The procedure automatically switches to a new audit table and archives the contents of the current table. In addition, this step involves setting the audit queue size and the suspend audit when device full configuration parameters.	"Setting up audit trail management" on page 449 For single-table auditing, "Single-table auditing" on page 458
3. Set up transaction log management in the sybsecurity database.	Determine how to handle the syslogs transaction log in the sybsecurity database, how to set the trunc log on chkpt database option and establishing a last-chance threshold procedure for syslogs if trunc log on chkpt is off.	"Setting up transaction log management" on page 456
4. Set auditing options.	Using sp_audit to establish the events to be audited.	"Setting global auditing options" on page 462
5. Enable auditing.	Using sp_configure to turn on the auditing configuration parameter. Adaptive Server begins writing audit records to the current audit table.	"Enabling and disabling auditing" on page 458.

Installing the audit system

The audit system is usually installed with auditinit, the Sybase installation program. Alternatively, you can install auditing without auditinit. For details, see "Installing auditing with installsecurity" on page 447. Installation and auditinit are discussed in the Adaptive Server installation and configuration documentation for your platform.

When you install auditing, you can establish the number of system tables you want to use for the audit trail, the device for each audit system table, and the device for the syslogs transaction log.

Tables and devices for the audit trail

You can specify up to eight system tables (sysaudits_01 through sysaudits_08). Plan to use at least two tables for the audit trail. Put each table on its own device separate from the master device. If you do this, you can use a threshold procedure to automatically archive the current audit table before it fills up and switch to a new empty table for the subsequent audit records.

Device for the *syslogs* transaction log table

When you install auditing, you must specify a separate device for the transaction log, which consists of the syslogs system table. The syslogs table, which exists in every database, contains a log of the transactions that are executed in the database.

Installing auditing with *installsecurity*

The *$SYBASE/scripts* directory contains installsecurity, a script for installing auditing.

Note This example assumes a server that uses a logical page size of 2K

To use installsecurity to install auditing:

1 Create the auditing devices and auditing database with the Transact-SQL disk init and create database commands. For example:

```
disk init name = "auditdev",
    physname = "/dev/dsk/c2d0s4",
    size = “10”
disk init name = "auditlogdev",
    physname = “/dev/dsk/c2d0s5”,
    size = “2M”
create database sybsecurity on auditdev
    log on auditlogdev
```

2 Use isql to execute the installsecurity script:

```
cd $SYBASE/scripts
setenv DSQUERY server_name
isql -Usa -Ppassword -Sserver_name <
installsecurity
```

3 Shut down and restart Adaptive Server.

When you have completed these steps, the sybsecurity database has one audit table (sysaudits_01) created on its own segment. You can enable auditing at this time, but should add more auditing tables with sp_addaudittable. For information about disk init, create database, and sp_addaudittable, see the Adaptive Server Reference Manual.

Moving the auditing database to multiple devices

Place the sybsecurity database on its own device, separate from the master database. If you have more than one audit table, place each table on its own device. It can also be helpful to put each table on a separate segment which points to a separate device. If you currently have sybsecurity on the same device as master, or if you want to move sybsecurity to another device, use one of the procedures described in the following sections. When you move the database, you can specify whether to save your existing global audit settings.

Moving *sybsecurity* without saving global audit settings

To move the sybsecurity database without saving the global audit settings:

1. Drop the sybsecurity database.
2. Install sybsecurity again using the installation procedure described in either:
   - The configuration documentation for your platform.
   - "Installing auditing with installsecurity" on page 447.
3. During the installation process, be sure to place the sybsecurity database on one or more devices, separate from the master device.

Moving *sybsecurity* and saving global audit settings

To move the sybsecurity database and save the global audit settings:

1. Dump the sybsecurity database.

   ```
   dump database sybsecurity to "/remote/sec_file"
   ```

2. Drop the sybsecurity database.

   ```
   drop database sybsecurity
   ```

3. Initialize the first device on which you want to place the sybsecurity database.

```
disk init name = "auditdev",
    physname = "/dev/dsk/c2d0s4",
    size = “10M”
```

4 Initialize the device where you want to place the security log.

```
disk init name = "auditlogdev",
    physname = "/dev/dsk/c2d0s5",
    size = “2M”
```

5 Create the new sybsecurity database.

```
create database sybsecurity on auditdev
    log on auditlogdev
```

6 Load the contents of the old sybsecurity database into the new database. The global audit settings are preserved.

```
load database sybsecurity from
"/remote/sec_file"
```

7 Run online database, which will upgrade sysaudits and sysauditoptions if necessary.

```
online database sybsecurity
```

8 Load the auditing system procedures using the configuration documentation for your platform.

To create more than one sysaudits table in sybsecurity:

1 Initialize the device where you want to place the additional table.

```
disk init name = "auditdev2",
    physname = "/dev/dsk/c2d0s6",
    size = “10M”
```

2 Extend the sybsecurity database to the device you initialized in step 1.

```
alter database sybsecurity on auditdev2 = “2M”
```

3 Run sp_addaudittable to create the next sysaudits table on the device you initialized in step 1.

```
sp_addaudittable auditdev2
```

4 Repeat steps 1–3 for each sysaudits table.

Setting up audit trail management

To effectively manage the audit trail:

1. Be sure that auditing is installed with two or more tables, each on a separate device. If not, consider adding additional audit tables and devices.
2. Write a threshold procedure and attach it to each audit table segment.
3. Set configuration parameters for the audit queue size and to indicate appropriate action should the current audit table become full.

The following sections assume that you have installed auditing with two or more tables, each on a separate device. If you have only one device for the audit tables, skip to "Single-table auditing" on page 458.

Setting up threshold procedures

Before enabling auditing, establish a threshold procedure to automatically switch auditing tables when the current table is full.

The threshold procedure for the audit device segments should:

- Make the next empty audit table current using sp_configure.
- Archive the audit table that is almost full using the insert and select commands.

Changing the current audit table

The current audit table configuration parameter establishes the table where Adaptive Server writes audit rows. As a System Security Officer, you can change the current audit table with sp_configure, using the following syntax:

```
sp_configure "current audit table", n
  [, "with truncate"]
```

where n is an integer that determines the new current audit table. The valid values for *n* are:

- 1 means sysaudits_01, 2 means sysaudits_02, and so forth.
- 0 tells Adaptive Server to automatically set the current audit table to the next table. For example, if your installation has three audit tables, sysaudits_01, sysaudits_02, and sysaudits_03, Adaptive Server sets the current audit table to:
  - 2 if the current audit table is sysaudits_01
  - 3 if the current audit table is sysaudits_02

- 1 if the current audit table is sysaudits_03

The with truncate option specifies that Adaptive Server should truncate the new table if it is not already empty. If you do not specify this option and the table is not empty, sp_configure fails.

Note If Adaptive Server truncates the current audit table and you have not archived the data, the table's audit records are lost. Archive the audit data before you use the with truncate option.

To execute sp_configure to change the current audit table, you must have the sso_role active. You can write a threshold procedure to automatically change the current audit table.

Archiving the audit table

You can use insert with select to copy the audit data into an existing table having the same columns as the audit tables in sybsecurity.

Be sure that the threshold procedure can successfully copy data into the archive table in another database:

1 Create the archive database on a separate device from the one containing audit tables in sybsecurity.

2 Create an archive table with columns identical to those in the sybsecurity audit tables. If such a table does not already exist, you can use select into to create an empty one by having a false condition in the where clause. For example:

```
use aud_db
go
select *
  into audit_data
  from sybsecurity.dbo.sysaudits_01
  where 1 = 2
```

The where condition is always false, so an empty duplicate of sysaudits_01 is created.

The select into/bulk copy database option must be turned on in the archive database (using sp_dboption) before you can use select into.

The threshold procedure, after using sp_configure to change the audit table, can use insert and select to copy data to the archive table in the archive database. The procedure can execute commands similar to these:

```
                    insert aud_db.sso_user.audit_data
                    select * from sybsecurity.dbo.sysaudits_01
```

Example threshold procedure for audit segments

This sample threshold procedure assumes that three tables are configured for auditing:

```
declare @audit_table_number int
/*
** Select the value of the current audit table
*/
select @audit_table_number = scc.value
from master.dbo.syscurconfigs scc, master.dbo.sysconfigures sc
where sc.config=scc.config and sc.name  = “current audit table”
/*
** Set the next audit table to be current.
** When the next audit table is specified as 0,
** the value is automatically set to the next one.
*/
exec sp_configure “current audit table”, 0, “with truncate”
/*
** Copy the audit records from the audit table
** that became full into another table.
*/
if @audit_table_number = 1
    begin
        insert aud_db.sso_user.sysaudits
            select * from sysaudits_01
        truncate table sysaudits_01
    end
else if @audit_table_number = 2
    begin
        insert aud_db.sso_user.sysaudits
            select * from sysaudits_02
        truncate table sysaudits_02
    end
return(0)
```

Attaching the threshold procedure to each audit segment

To attach the threshold procedure to each audit table segment, use the sp_addthreshold.

Before executing sp_addthreshold:

- Determine the number of audit tables configured for your installation and the names of their device segments
- Have the permissions and roles you need for sp_addthreshold for all the commands in the threshold procedure

Warning! sp_addthreshold and sp_modifythreshold check to ensure that only a user with sa_role directly granted can add or modify a threshold. All system-defined roles that are active when you add or modify a threshold are inserted as valid roles for your login in the systhresholds table. However, only directly granted roles are activated when the threshold procedure fires.

Audit tables and their segments

When you install auditing, auditinit displays the name of each audit table and its segment. The segment names are "aud_seg1" for sysaudits_01, "aud_seg2" for sysaudits_02, and so forth. You can find information about the segments in the sybsecurity database if you execute sp_helpsegment with sybsecurity as your current database. One way to find the number of audit tables for your installation is to execute the following SQL commands:

```
use sybsecurity
go
select count(*) from sysobjects
  where name like "sysaudit%"
go
```

In addition, you can get information about the audit tables and the sybsecurity database by executing the following SQL commands:

```
sp_helpdb sybsecurity
go
use sybsecurity
go
sp_help sysaudits_01
go
sp_help sysaudits_02
go
  ...
```

Required roles and permissions

To execute sp_addthreshold, you must be either the Database Owner or a System Administrator. A System Security Officer should be the owner of the sybsecurity database and, therefore, should be able to execute sp_addthreshold. In addition to being able to execute sp_addthreshold, you must have permission to execute all the commands in your threshold procedure. For example, to execute sp_configure for current audit table, the sso_role must be active. When the threshold procedure fires, Adaptive Server attempts to turn on all the roles and permissions that were in effect when you executed sp_addthreshold.

To attach the threshold procedure audit_thresh to three device segments:

```
use sybsecurity
go
sp_addthreshold sybsecurity, aud_seg_01, 250, audit_thresh
sp_addthreshold sybsecurity, aud_seg_02, 250, audit_thresh
sp_addthreshold sybsecurity, aud_seg_03, 250, audit_thresh
go
```

The sample threshold procedure audit_thresh receives control when fewer than 250 free pages remain in the current audit table.

For more information about adding threshold procedures, see Chapter 29, "Managing Free Space with Thresholds."

Auditing with the sample threshold procedure in place

After you enable auditing, Adaptive Server writes all audit data to the initial current audit table, sysaudits_01. When sysaudits_01 is within 250 pages of being full, the threshold procedure audit_thresh fires. The procedure switches the current audit table to sysaudits_02, and, immediately, Adaptive Server starts writing new audit records to sysaudits_02. The procedure also copies all audit data from sysaudits_01 to the audit_data archive table in the audit_db database. The rotation of the audit tables continues in this fashion without manual intervention.

Setting auditing configuration parameters

Set the following configuration parameters for your auditing installation:

- audit queue size sets the number of records in the audit queue in memory.

- suspend audit when device full determines what Adaptive Server does if the current audit table becomes completely full. The full condition occurs only if the threshold procedure attached to the current table segment is not functioning properly.

Setting the size of the audit queue

The memory requirement for a single audit record is 424 bytes. The default size for the audit queue is 100 records, which requires approximately 42K.

To set the size of the audit queue, use sp_configure. The syntax is:

```
sp_configure "audit queue size", [value]
```

value is the number of records that the audit queue can hold. The minimum value is 1, and the maximum is 65,535. For example, to set the audit queue size to 300, execute:

```
sp_configure "audit queue size", 300
```

For more information about setting the audit queue size and other configuration parameters, see Chapter 5, “Setting Configuration Parameters.”

Suspending auditing if devices are full

If you have two or more audit tables, each on a separate device other than the master device, and have a threshold procedure for each audit table segment, the audit devices should never become full. Only if a threshold procedure is not functioning properly would the “full” condition occur. You can use sp_configure to set the suspend audit when device full parameter to determine what happens if the devices do become full. Choose one of these options:

- Suspend the auditing process and all user processes that cause an auditable event. Resume normal operation after a System Security Officer clears the current audit table.
- Truncate the next audit table and start using it. This allows normal operation to proceed without intervention from a System Security Officer.

To set this configuration parameter, use sp_configure. You must have the sso_role active. The syntax is:

```
sp_configure "suspend audit when device full",
    [0|1]
```

0 truncates the next audit table and starts using it as the current audit table whenever the current audit table becomes full. If you set the parameter to 0, the audit process is never suspended; however, older audit records will be lost if they have not been archived.

1 (the default value) suspends the audit process and all user processes that cause an auditable event. To resume normal operation, the System Security Officer must log in and set up an empty table as the current audit table. During this period, the System Security Officer is exempt from normal auditing. If the System Security Officer's actions would generate audit records under normal operation, Adaptive Server sends an error message and information about the event to the error log.

If you have a threshold procedure attached to the audit table segments, set suspend audit when device full to 1 (on). If it is set to 0 (off), Adaptive Server may truncate the audit table that is full before your threshold procedure has a chance to archive your audit records.

Setting up transaction log management

This section describes guidelines for managing the transaction log in sybsecurity.

If the trunc log on chkpt database option is active, Adaptive Server truncates syslogs every time it performs an automatic checkpoint. After auditing is installed, the value of trunc log on chkpt is on, but you can use sp_dboption to change its value.

Truncating the transaction log

If you enable the trunc log on chkpt option for the sybsecurity database, you do not need to worry about the transaction log becoming full. Adaptive Server truncates the log whenever it performs a checkpoint. With this option on, you cannot use dump transaction to dump the transaction log, but you can use dump database to dump the database.

If you follow the procedures in "Setting up threshold procedures" on page 450, audit tables are automatically archived to tables in another database. You can use standard backup and recovery procedures for this archive database.

If a crash occurs on the sybsecurity device, you can reload the database and resume auditing. At most, only the records in the in-memory audit queue and the current audit table are lost because the archive database contains all other audit data. After you reload the database, use sp_configure with truncate to set and truncate the current audit table.

If you have not changed server-wide auditing options since you dumped the database, all auditing options stored in sysauditoptions are automatically restored when you reload sybsecurity. If not, you can run a script to set the options prior to resuming auditing.

Managing the transaction log with no truncation

If you use db_option to turn the trunc log on chkpt off, the transaction log may fill up. Plan to attach a *last-chance threshold procedure* to the transaction log segment. This procedure gets control when the amount of space remaining on the segment is less than a threshold amount computed automatically by Adaptive Server. The threshold amount is an estimate of the number of free log pages that would be required to back up the transaction log.

The default name of the last-chance threshold procedure is sp_thresholdaction, but you can specify a different name with sp_modifythreshold, as long as you have the sa_role active.

Note sp_modifythreshold checks to ensure you have "sa_role" active. See "Attaching the threshold procedure to each audit segment" on page 452 for more information.

Adaptive Server does not supply a default procedure, but Chapter 29, "Managing Free Space with Thresholds" contains examples of last-chance threshold procedures. The procedure should execute the dump transaction command, which truncates the log. When the transaction log reaches the last-chance threshold point, any transaction that is running is suspended until space is available. The suspension occurs because the option abort xact when log is full is always set to FALSE for the sybsecurity database. You cannot change this option.

With the trunc log on chkpt option off, you can use standard backup and recovery procedures for the sybsecurity database, but be aware that the audit tables in the restored database may not be in sync with their status at the time of a device failure.

Enabling and disabling auditing

To enable or disable auditing, use sp_configure with the auditing configuration parameter. The syntax is:

```
sp_configure "auditing", [0 | 1 ]
```

1 enables auditing. 0 disables auditing. For example, to enable auditing, enter:

```
sp_configure "auditing", 1
```

Note When you enable or disable auditing, Adaptive Server automatically generates an audit record. See event codes 73 and 74 in Table 12-6 on page 473.

Single-table auditing

Sybase strongly recommends that you *not* use single-device auditing for production systems. If you use only a single audit table, you create a window of time while you are archiving audit data and truncating the audit table during which incoming audit records will be lost. There is no way to avoid this when using only a single audit table.

If you use only a single audit table, your audit table is likely to fill up. The consequences of this depend on how you have set suspend audit when device full. If you have suspend audit when device full set to on, the audit process is suspended, as are all user processes that cause auditable events. If suspend audit when device full is off, the audit table is truncated, and you lose all the audit records that were in the audit table.

For *non-production* systems, where the loss of a small number of audit records may be acceptable, you can use a single table for auditing, if you cannot spare the additional disk space for multiple audit tables, or you do not have additional devices to use.

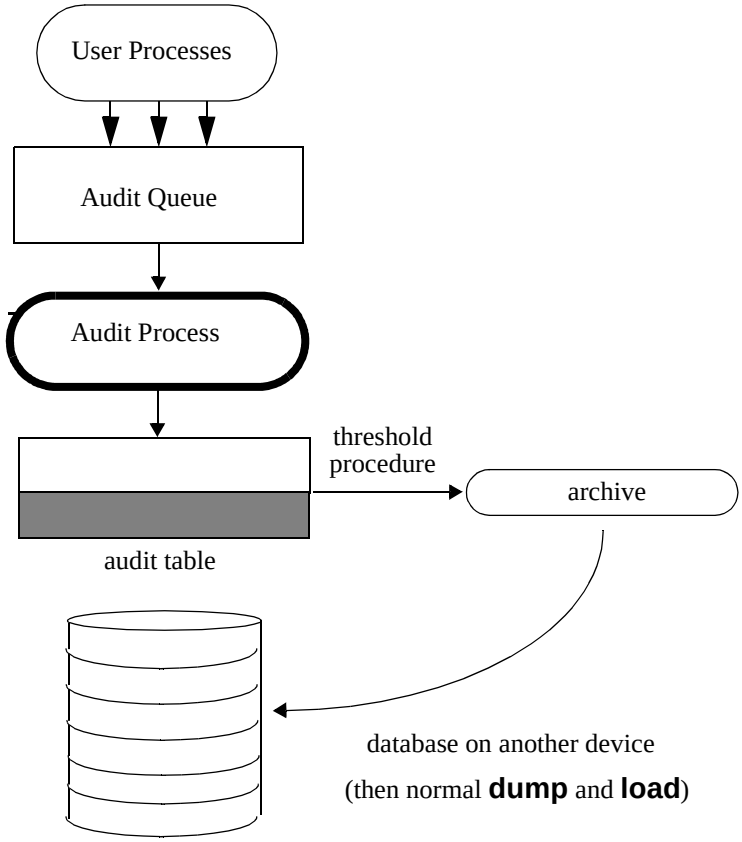
The procedure for using a single audit table is similar to using multiple audit tables, with these exceptions:

- During installation, you specify only one system table to use for auditing.
- During installation, you specify only one device for the audit system table.

- The threshold procedure you create for archiving audit records is different from the one you would create if you were using multiple audit tables.

Figure 12-2 shows how the auditing process works with a single audit table.

Figure 12-2: Auditing with a single audit table



Establishing and managing single-table auditing

Table 12-2 provides an overview of managing single-table auditing.

Table 12-2: Auditing process for single-table auditing

Action	Description	See
1. Install auditing.	Installation of auditing, which involves setting the number of audit tables and assigning devices for the audit trail and the syslogs transaction log in the sybsecurity database.	The the installation documentation for your platform
2. Set up the audit process to manage the audit trail.	Writing and establishing a threshold procedure that receives control when the audit table is nearly full. The procedure automatically writes the contents of the audit table to another table, and then truncates the audit table. In addition, this step involves setting the audit queue size and suspend audit when device full configuration parameters.	"Establishing and managing single-table auditing" on page 459. "Threshold procedure for single-table auditing" on page 461.
3. Set up the audit process to manage the syslogs transaction log in the sybsecurity database.	Determining how to handle the syslogs transaction log in the sybsecurity database. The task includes determining the setting of the trunc log on chkpt database option and establishing a last-chance threshold procedure for syslogs if trunc log on chkpt is off.	"Setting up transaction log management" on page 456.
4. Set auditing options.	Using sp_audit to establish the events to be audited. **Note** No audit records are generated until auditing is turned on with sp_configure.	"Setting global auditing options" on page 462
5. Enable auditing.	Using sp_configure to turn on the auditing configuration parameter. Adaptive Server begins writing audit records for audited events to the current audit table.	"Enabling and disabling auditing" on page 458

Threshold procedure for single-table auditing

For single-table auditing, the threshold procedure should:

- Archive the almost-full audit table to another table, using the insert and select commands.
- Truncate the audit table to create space for new audit records, using the truncate table command.

Before you can archive your audit records, create an archive table that has the same columns as your audit table. After you have done this, your threshold procedure can use insert with select to copy the audit records into the archive table.

Here is a sample threshold procedure for use with a single audit table:

```
create procedure audit_thresh as
/*
** copy the audit records from the audit table to
** the archive table
*/
insert aud_db.sso_user.audit_data
    select * from sysaudits_01
return(0)
go
/*
** truncate the audit table to make room for new
** audit records
*/
truncate table “sysaudits_01”
go
```

After you have created your threshold procedure, you will need to attach the procedure to the audit table segment. For instructions, see “Attaching the threshold procedure to each audit segment” on page 452.

Warning! On a multiprocessor, the audit table may fill up even if you have a threshold procedure that triggers before the audit table is full. For example, if the threshold procedure is running on a heavily loaded CPU, and a user process performing auditable events is running on a less heavily loaded CPU, it is possible that the audit table can fill up before the threshold procedure triggers. The configuration parameter suspend audit when device full determines what happens when the audit table fills up. For information about setting this parameter, see “Suspending auditing if devices are full” on page 455.

What happens when the current audit table is full?

When the current audit table is full:

1. The audit process attempts to insert the next audit record into the table. This fails, so the audit process terminates. An error message goes to the error log.
2. When a user attempts to perform an auditable event, the event cannot be completed because auditing cannot proceed. The user process terminates. Users who do not attempt to perform an auditable event are unaffected.
3. If you have login auditing enabled, no one can log in to the server except a System Security Officer.
4. If you are auditing commands executed with the sso_role active, the System Security Officer will be unable to execute commands.

Recovering when the current audit table is full

If the current audit device and the audit queue becomes full, the System Security Officer becomes exempt from auditing. Every auditable event performed by a System Security Officer after this point sends a warning message to the error log file. The message states the date and time and a warning that an audit has been missed, as well as the login name, event code, and other information that would normally be stored in the extrainfo column of the audit table.

When the current audit table is full, the System Security Officer can archive and truncate the audit table as described in "Archiving the audit table" on page 451. A System Administrator can execute shutdown to stop the server and then restart the server to reestablish auditing.

If the audit system terminates abnormally, the System Security Officer can shut down the server after the current audit table has been archived and truncated. Normally, only the System Administrator can execute shutdown.

Setting global auditing options

After you have installed auditing, you can use sp_audit to set auditing options. The syntax for sp_audit is:

sp_audit *option*, *login_name*, *object_name* [,*setting*]

If you run sp_audit with no parameters, it provides a complete list of the options. For details about sp_audit, see the Adaptive Server Reference Manual.

Note No auditing occurs until you activate auditing for the server. For information on how to start auditing, see "Enabling and disabling auditing" on page 458.

Auditing options: Their types and requirements

The values you can specify for the *login_name* and *object_name* parameters to sp_audit depend on the type of auditing option you specify:

- Global options apply to commands that affect the entire server, such as booting the server, disk commands, and allowing ad hoc, user-defined audit records. Option settings for global events are stored in the sybsecurity..sysauditoptions system table.
- Database-specific options apply to a database. Examples include altering a database, bulk copy (bcp in) of data into a database, granting or revoking access to objects in a database, and creating objects in a database. Option settings for database-specific events are stored in the master..sysdatabases system table.
- Object-specific options apply to a specific object. Examples include selecting, inserting, updating, or deleting rows of a particular table or view and the execution of a particular trigger or procedure. Option settings for object-specific events are stored in the sysobjects system table in the relevant database.
- User-specific options apply to a specific user or system role. Examples include accesses by a particular user to any table or view or all actions performed when a particular system role, such as sa_role, is active. Option settings for individual users are stored in master..syslogins. The settings for system roles are stored in master..sysauditoptions.

Table 12-3 shows:

- Valid values for the option and the type of each option – global, database-specific, object-specific, or user-specific

- Valid values for the *login_name* and *object_name* parameters for each option
- The database to be in when you set the auditing option
- The command or access that is audited when you set the option
- An example for each option

The default value of all options is off.

Table 12-3: Auditing options, requirements, and examples

Option (Option type)	*login_name*	*object_name*	Database to be in to set the option	Command or access being audited
adhoc (user-specific)	all	all	Any	Allows users to use sp_addauditrecord
Example:	`sp_audit "adhoc", "all", "all", "on"`			
	(Enables ad hoc user-defined auditing records.)			
all (user-specific)	A login name or role	all	Any	All actions of a particular user or by users with a particular role active
Example:	`sp_audit "all", "sa_role", "all", "on"`			
	(Turns auditing on for all actions in which the sa_role is active.)			
alter (database-specific)	all	Database to be audited	Any	alter database, alter table
Example:	`sp_audit @option = "alter", @login_name = "all", @object_name = "master", @setting = "on"`			
	(Turns auditing on for all executions of alter database and alter table in the master database.)			
bcp(database-specific)	all	Database to be audited	Any	bcp in
Example:	`sp_audit "bcp", "all", "pubs2"`			
	(Returns the status of bcp auditing in the pubs2 database. If you do not specify a value for *setting*, Adaptive Server returns the status of auditing for the option you specify)			
bind(database-specific)	all	Database to be audited	Any	sp_bindefault, sp_bindmsg, sp_bindrule
Example:	`sp_audit "bind", "all", "planning", "off"`			
	(Turns bind auditing off for the planning database.)			

Option (Option type)	*login_name*	*object_name*	Database to be in to set the option	Command or access being audited
cmdtext(user-specific)	A login name or a role	all	Any	All actions of a particular user or by users with a particular role active. (Does not reflect whether or not the text in question passed permission checks or not. *eventmod* always has a value of 1.)
Example:	`sp_audit "cmdtext", "dbo", "off"` (Turns text auditing off for Database Owners.)			
create(database-specific)	all	Database to be audited Specify master for *object_name* if you want to audit create database. You will also be auditing the creation of other objects in master.	Any	create database, create table, create procedure, create trigger, create rule, create default, sp_addmessage, create view
Example:	`sp_audit "create", "all", "planning", "pass"` (Turns on auditing of successful object creations in the planning database. The current status of auditing create database is not affected because you did not specify the master database.)			
dbaccess(database-specific)	all	Database to be audited	Any	Any access to the database from another database
Example:	`sp_audit "dbaccess", "all", "project", "on"` (Audits all external accesses to the project database.)			
dbcc(global)	all	all	Any	dbcc
Example:	`sp_audit "dbcc", "all", "all", "on"` (Audits all executions of the dbcc command.)			
delete(object-specific)	all	Table or view, default table, or default view	The database of the table or view (except tempdb)	delete from a table, delete from a view
Example:	`sp_audit "delete", "all", "default table", "on"` (Audits all delete actions for all future tables in the current database.)			
disk(global)	all	all	Any	disk init, disk refit, disk reinit, disk mirror, disk unmirror, disk remirror

Option (Option type)	*login_name*	*object_name*	Database to be in to set the option	Command or access being audited
Example:	`sp_audit "disk", "all", "all", "on"`			
	(Audits all disk actions for the server.)			
drop(database-specific)	all	Database to be audited	Any	drop database, drop table, drop procedure, drop trigger, drop rule, drop default, sp_dropmessage, drop view
Example:	`sp_audit "drop", "all", "financial", "fail"`			
	(Audits all drop commands in the financial database that fail permission checks.)			
dump(database-specific)	all	Database to be audited	Any	dump database, dump transaction
Example:	`sp_audit "dump", "all", "pubs2", "on"`			
	(Audits dump commands in the pubs2 database.)			
errors(global)	all	all	Any	Fatal error, non-fatal error
Example:	`sp_audit "errors", "all", "all", "on"`			
	(Audits errors throughout the server.)			
exec_procedure(object-specific)	all	Procedure or default procedure	The database of the procedure (except tempdb)	execute
Example:	`sp_audit "exec_procedure", "all", "default procedure", "off"`			
	(Turns automatic auditing off of new procedures in the current database.)			
exec_trigger(object-specific)	all	Trigger or default trigger	The database of the trigger (except tempdb)	Any command that fires the trigger
Example:	`sp_audit "exec_trigger", "all", "trig_fix_plan", "fail"`			
	(Audits all failed executions of the trig_fix_plan trigger in the current database.)			
func_dbaccess(database-specific)	all	Database	Any	Access to the database via Transact-SQL built-in functions
Example:	`sp_audit @option="func_dbaccess", @login_name="all", @object_name = "strategy", @setting = "on"`			
	(Audits accesses to the strategy database via built-in functions.)			
func_obj_access(object-specific)	all	Object	Any	Access to an object via Transact-SQL built-in functions

Option (Option type)	*login_name*	*object_name*	Database to be in to set the option	Command or access being audited
Example:	`sp_audit @option="func_obj_access", @login_name="all", @object_name = "customer", @setting = "on"`			
	(Audits accesses to the customer table via built-in functions.)			
grant(database-specific)	all	Database to be audited	Any	grant
Example:	`sp_audit @option="grant", @login_name="all", @object_name = "planning", @setting = "on"`			
	(Audits all grants in the planning database.)			
insert(object-specific)	all	Table or view, default table, or default view	The database of the object (except tempdb)	insert into a table, insert into a view
Example:	`sp_audit "insert", "all", "dpt_101_view", "on"`			
	(Audits all inserts into the dpt_101_view view in the current database..)			
load(database-specific)	all	Database to be audited	Any	load database, load transaction
Example:	`sp_audit "load", "all", "projects_db", "fail"`			
	(Audits all failed executions of database and transaction loads in the projects_db database.)			
login(global)	all	all	Any	Any login to Adaptive Server
Example:	`sp_audit "login", "all", "all", "fail"`			
	(Audits all failed attempts to log in to the server.)			
logout(global)	all	all	Any	Any logout from Adaptive Server
Example:	`sp_audit "logout", "all", "all", "off"`			
	(Turns auditing off of logouts from the server.)			
reference(object-specific)	all	Table to be audited	Any	Creation of a reference between tables
Example:	`sp_audit "reference", "all", "titles", "off"`			
	(Turns off auditing of the creation of references between the titles table and other tables.)			
revoke(database-specific)	all	Database to be audited	Any	revoke
Example:	`sp_audit "revoke", "all", "payments_db", "off"`			
	(Turns off auditing of the execution of revoke in the payments_db database.)			
rpc(global)	all	all	Any	Remote procedure calls (either in or out)
Example:	`sp_audit "rpc", "all", "all", "on"`			
	(Audits all remote procedure calls out of or into the server.)			

Option (Option type)	*login_name*	*object_name*	Database to be in to set the option	Command or access being audited
security(global)	all	all	Any	Server-wide security-relevant events. See the "security" option in Table 12-3.
Example:	`sp_audit "security", "all", "all", "on"`			
	(Audits server-wide security-relevant events in the server.)			
select(object-specific)	all	Table or view, default table, or default view	The database of the object (except tempdb)	select from a table, select from a view
Example:	`sp_audit "select", "all", "customer", "fail"`			
	(Audits all failed selects from the customer table in the current database.)			
setuser(database-specific)	all	all	Any	setuser
Example:	`sp_audit "setuser", "all", "projdb", "on"`			
	(Audits all executions of setuser in the projdb database.)			
table_access(user-specific)	Login name	all	Any	select, delete, update, or insert access in a table
Example:	`sp_audit "table_access", "smithson", "all", "on"`			
	(Audits all table accesses by the login named "smithson".)			
truncate(database-specific)	all	Database to be audited	Any	truncate table
Example:	`sp_audit "truncate", "all", "customer", "on"`			
	(Audits all table truncations in the customer database.)			
unbind(database-specific)	all	Database to be audited	Any	sp_unbindefault, sp_unbindrule, sp_unbindmsg
Example:	`sp_audit "unbind", "all", "master", "fail"`			
	(Audits all failed attempts of unbinding in the master database.)			
update(object-specific)	all	View, default table, or default view	The database of the object (except tempdb)	update to a table, update to a view
Example:	`sp_audit "update", "all", "projects", "on"`			
	(Audits all attempts by users to update the projects table in the current database.)			
view_access(user-specific)	Login name	all	Any	select, delete, insert, or update to a view
Example:	`sp_audit "view_access", "joe", "all", "off"`			
	(Turns off view auditing of user "joe".)			

Examples of setting auditing options

Suppose you want to audit all failed deletions on the projects table in the company_operations database and for all new tables in the database. Use the object-specific delete option for the projects table and use default table for all future tables in the database. To set object-specific auditing options, you must be in the object's database before you execute sp_audit:

```
sp_audit "security", "all", "all", "fail"
```

- For this example, execute:

```
use company_operations
go
sp_audit "delete", "all", "projects", "fail"
go
sp_audit "delete", "all", "default table",
"fail"
go
```

Determining current auditing settings

To determine the current auditing settings for a given option, use sp_displayaudit. The syntax is:

sp_displayaudit [*procedure* | *object* | *login* | *database* | *global* | *default_object* | *default_procedure* [, *name*]]

For more information, see sp_displayaudit in the Adaptive Server Reference Manual.

Adding user-specified records to the audit trail

sp_addauditrecord allows users to enter comments into the audit trail. The syntax is:

sp_addauditrecord [*text*] [, *db_name*] [, *obj name*] [, *owner_name*] [, *dbid*] [, *objid*]

All the parameters are optional.

- *text* is the text of the message that you want to add to the extrainfo audit table.
- *db_name* is the name of the database referred to in the record, which is inserted into the dbname column of the current audit table.

- *obj_name* is the name of the object referred to in the record, which is inserted into the objname column of the current audit table.
- *owner_name* is the owner of the object referred to in the record, which is inserted into the objowner column of the current audit table.
- *dbid* is an integer value representing the database ID number of db_name, which is inserted into the dbid column of the current audit table. Do not place it in quotes.
- *objid* is an integer value representing the object ID number of obj_name. Do not place it in quotes. *objid* is inserted into the objid column of the current audit table.

You can use sp_addauditrecord if:

- You have execute permission on sp_addauditrecord.
- The auditing configuration parameter was activated with sp_configure.
- The adhoc auditing option was enabled with sp_audit.

By default, only a System Security Officer and the Database Owner of sybsecurity can use sp_addauditrecord. Permission to execute it may be granted to other users.

Examples of adding user-defined audit records

The following example adds a record to the current audit table. The text portion is entered into the extrainfo column of the current audit table, “corporate” into the dbname column, “payroll” into the objname column, “dbo” into the objowner column, “10” into the dbid column, and “1004738270” into the objid column:

```
sp_addauditrecord "I gave A. Smith permission to
view the payroll table in the corporate database.
This permission was in effect from 3:10 to 3:30 pm
on 9/22/92.", "corporate", "payroll", "dbo", 10,
1004738270
```

The following example inserts information only into the extrainfo and dbname columns of the current audit table:

```
sp_addauditrecord @text="I am disabling auditing
briefly while we reconfigure the system",
@db_name="corporate"
```

Querying the audit trail

To query the audit trail, use SQL to select and summarize the audit data. If you follow the procedures discussed in "Setting up audit trail management" on page 449, the audit data is automatically archived to one or more tables in another database. For example, assume that the audit data resides in a table called audit_data in the audit_db database. To select audit records for tasks performed by "bob" on July 5, 1993, execute:

```
use audit_db
go
select * from audit_data
  where loginname = "bob"
  and eventtime like "Jul 5% 93"
go
```

This command requests audit records for commands performed in the pubs2 database by users with the System Security Officer role active:

```
select * from audit_data
  where extrainfo like "%sso_role%"
  and dbname = "pubs2"
go
```

This command requests audit records for all table truncations (event 64):

```
select * from audit_data
  where event = 64
go
```

Understanding the audit tables

The system audit tables can be accessed only by a System Security Officer, who can read the tables by executing SQL commands. The only commands that are allowed on the system audit tables are select and truncate.

Table 12-4 describes the columns in all audit tables.

Table 12-4: Columns in each audit table

Column name	Datatype	Description
event	smallint	Type of event being audited. See Table 12-6 on page 473.

Column name	Datatype	Description
eventmod	smallint	More information about the event being audited. indicates whether or not the event in question passed permission checks. Possible values are: • 0 = no modifier for this event • 1 = the event passed permission checking • 2 = the event failed permission checking
spid	smallint	ID of the process that caused the audit record to be written.
eventtime	datetime	Date and time that the audited event occurred.
sequence	smallint	Sequence number of the record within a single event. Some events require more than one audit record.
suid	smallint	Server login ID of the user who performed the audited event.
dbid	int null	Database ID in which the audited event occurred, or in which the object, stored procedure, or trigger resides, depending on the type of event.
objid	int null	ID of the accessed object, stored procedure, or trigger.
xactid	binary(6) null	ID of the transaction containing the audited event. For a multi-database transaction, this is the transaction ID from the database where the transaction originated.
loginname	varchar(30) null	Login name corresponding to the suid.
dbname	varchar(30) null	Database name corresponding to the dbid.
objname	varchar(30) null	Object name corresponding to the objid.
objowner	varchar(30) null	Name of the owner of objid.
extrainfo	varchar(255) null	Additional information about the audited event. This column contains a sequence of items separated by semicolons. For details, see "Reading the extrainfo column" on page 472.

Reading the *extrainfo* column

The extrainfo column contains a sequence of data separated by semicolons. The data is organized in the following categories.

Table 12-5: Information in the extrainfo column

Position	Category	Description
1	Roles	A list of active roles, separated by blanks.
2	Keywords or Options	The name of the keyword or option that was used for the event. For example, for the alter table command, the add column or drop constraint options might have been used. If multiple keywords or options are listed, they are separated by commas.

Position	Category	Description
3	Previous value	If the event resulted in the update of a value, this item contains the value prior to the update.
4	Current value	If the event resulted in the update of a value, this item contains the new value.
5	Other information	Additional security-relevant information that is recorded for the event.
6	Proxy information	The original login name if the event occurred while a set proxy was in effect.
7	Principal name	The principal name from the underlying security mechanism if the user's login is the secure default login, and the user logged into Adaptive Server via unified login. The value of this item is NULL if the secure default login is not being used.

This example shows an extrainfo column entry for the event of changing an auditing configuration parameter.

```
sso_role;suspend audit when device full;1;0;;ralph;
```

This entry indicates that a System Security Officer changed suspend audit when device full from 1 to 0. There is no "other information" for this entry. The sixth category indicates that the user "ralph" was operating with a proxy login. No principal name is provided.

The other fields in the audit record give other pertinent information. For example, the record contains the server user ID (suid) and the login name (loginname).

Table 12-6 lists the values that appear in the event column, arranged by sp_audit option. The "Information in extrainfo" column describes information that might appear in the extrainfo column of an audit table, based on the categories described in Table 12-5.

Table 12-6: Values in event and extrainfo columns

Audit Option	Command or access to be audited	event	Information in extrainfo
(Automatically audited event not controlled by an option)	Enabling auditing with: sp_configure auditing	73	-
(Automatically audited event not controlled by an option)	Disabling auditing with: sp_configure auditing	74	-
adhoc	User-defined audit record	1	extrainfo is filled by the *text* parameter of sp_addauditrecord

Audit Option	Command or access to be audited	event	Information in extrainfo
alter	alter database	2	*Keywords or options:* alter maxhold alter size
	alter table	3	*Keywords or options:* add column drop column replace column add constraint drop constraint
bcp	bcp in	4	-
bind	sp_bindefault	6	*Other information:* Name of the default
	sp_bindmsg	7	*Other information:* Message ID
	sp_bindrule	8	*Other information:* Name of the rule
create	create database	9	-
	create default	14	-
	create procedure	11	-
	create rule	13	-
	create table	10	-
	create trigger	12	-
	create view	16	-
	sp_addmessage	15	*Other information:* Message number
dbaccess	Any access to the database by any user	17	*Keywords or options:* use cmd outside reference
dbcc	dbcc (all keywords)	81	*Keywords or options:* Any of the dbcc keywords such as checkstorage and the options for that keyword.
delete	delete from a table	18	*Keywords or options:* delete
	delete from a view	19	*Keywords or options:* delete

Audit Option	Command or access to be audited	event	Information in extrainfo
disk	disk init	20	*Keywords or options:* disk init *Other information:* Name of the disk
	disk mirror	23	*Keywords or options:* disk mirror *Other information:* Name of the disk
	disk refit	21	*Keywords or options:* disk refit *Other information:* Name of the disk
	disk reinit	22	*Keywords or options:* disk reinit *Other information:* Name of the disk
	disk remirror	25	*Keywords or options:* disk remirror *Other information:* Name of the disk
	disk unmirror	24	*Keywords or options:* disk unmirror *Other information:* Name of the disk
drop	drop database	26	-
	drop default	31	-
	drop procedure	28	-
	drop table	27	-
	drop trigger	29	-
	drop rule	30	-
	drop view	33	-
	sp_dropmessage	32	*Other information:* Message number
dump	dump database	34	-
	dump transaction	35	-
errors	Fatal error	36	*Other information:* *Error number*.*Severity*.*State*
	Non-fatal error	37	*Other information:* *Error number*.*Severity*.*State*
exec_procedure	Execution of a procedure	38	*Other information:* All input parameters
exec_trigger	Execution of a trigger	39	-
func_obj_access, func_dbaccess	Accesses to objects and databases via Transact-SQL functions	85	-
grant	grant	40	-

Audit Option	Command or access to be audited	event	Information in extrainfo
insert	insert into a table	41	*Keywords or options:* If insert is used: insert If select into is used: insert into followed by the fully qualified object name
	insert into a view	42	*Keywords or options*: insert
load	load database	43	-
	load transaction	44	-
login	Any login to the server	45	*Other information:* Host name of the machine from which login was done
logout	Any logouts from the server	46	*Other information:* Host name of the machine from which login was done
reference	Creation of references to tables	91	*Keywords or options*: reference *Other information:* Name of the referencing table
revoke	revoke	47	-
rpc	Remote procedure call from another server	48	*Keywords or options:* Name of client program *Other information:* Server name, host name of the machine from which the RPC was done.
	Remote procedure call to another server	49	*Keywords or options:* Procedure name
security	connect to (CIS only)	90	*Keywords or options:* connect to
	kill (CIS only)	89	*Keywords or options:* kill
	online database	83	-
	proc_role function (executed from within a system procedure)	80	*Other information:* Required roles
	Regeneration of a password by an SSO	76	*Keywords or options:* Setting SSO password *Other information:* Login name
	Role toggling	55	*Previous value:* on or off *Current value:* on or off *Other information:* Name of the role being set

Audit Option	Command or access to be audited	event	Information in extrainfo
	Server boot	50	*Other information:* -d*masterdevicename* -i*interfaces file path* -S*servername* -e*errorfilename*
	Server shutdown	51	*Keywords or options:* shutdown
	set proxy or set session authorization	88	*Previous value:* Previous suid *Current value:* New suid
	sp_configure	82	*Other information:* • If a parameter is being set: number of configuration parameter • If a configuration file is being used to set parameters: name of the configuration file
	valid_user	85	*Keywords or options:* valid_user
select	select from a table	62	*Keywords or options:* select into select readtext
	select from a view	63	*Keywords or options:* select into select readtext
setuser	setuser	84	*Other information:* Name of the user being set
table_access	delete	18	*Keywords or options:* delete
	insert	41	*Keywords or options:* insert
	select	62	*Keywords or options:* select into select readtext
	update	70	*Keywords or options:* update writetext
truncate	truncate table	64	-

Audit Option	Command or access to be audited	event	Information in extrainfo
unbind	sp_unbindefault	67	-
	sp_unbindmsg	69	-
	sp_unbindrule	68	-
update	update to a table	70	*Keywords or options:* update writetext
	update to a view	71	*Keywords or options:* update writetext
view_access	delete	19	*Keywords or options:* delete
	insert	42	*Keywords or options:* insert
	select	63	*Keywords or options:* select into select readtext
	update	71	*Keywords or options:* update writetext

CHAPTER 13

Managing Remote Servers

This chapter discusses the steps the System Administrator and System Security Officer of each Adaptive Server must execute to enable **remote procedure calls** (RPCs).

Topics covered in this chapter include:

Overview

Users on a local Adaptive Server can execute stored procedures on a remote Adaptive Server. Executing an RPC sends the results of the remote process to the calling process—usually displayed on the user's screen.

Note The use of remote servers is not included in the evaluated configuration.

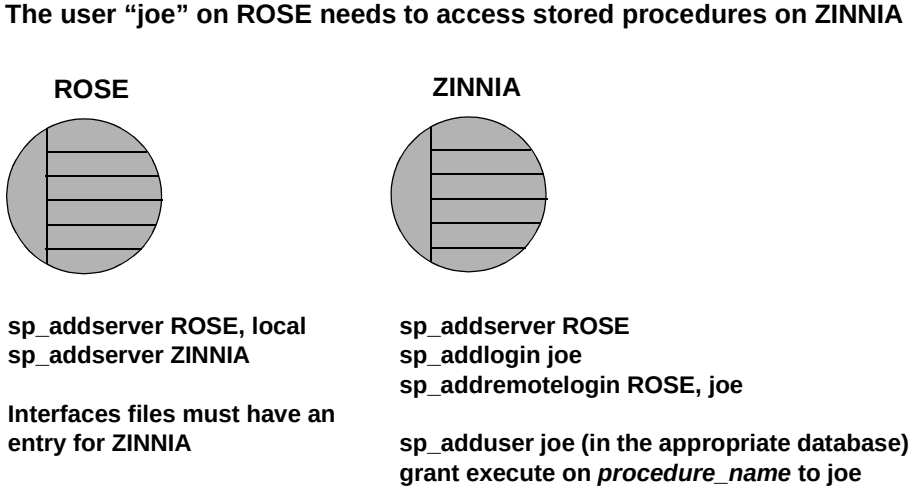
To enable RPCs, the System Administrator and System Security Officer of each Adaptive Server must execute the following steps:

- On the local server:
  - (System Security Officer) Use sp_addserver to list the local server and remote server in the system table master..sysservers.
  - List the remote server in the interfaces file or Directory Service for the local server.

    - Reboot the local server so the global variable @@servername is set to the name of the local server. If this variable is not set properly, users cannot execute RPCs from the local server on any remote server.
- On the remote server:
    - (System Security Officer) Use sp_addserver to list the server originating the RPC in the system table master..sysservers.
    - To allow the user who is originating the remote procedure access to the server, a System Security Officer uses sp_addlogin, and a System Administrator uses sp_addremotelogin.
    - Add the remote login name as a user of the appropriate database and grant that login permission to execute the procedure. (If execute permission is granted to "public", the user does not need to be granted specific permission.)

Figure 13-1 shows how to set up servers for remote access.

Figure 13-1: Setting up servers to allow remote procedure calls



For operating-system-specific information about handling remote servers, see the the installation documentation for your platform.

Managing remote servers

Table 13-1 lists the tasks related to managing remote servers and the system procedures you use to perform the tasks.

Table 13-1: Tasks related to managing remote servers

To	Use	See
Add a remote server	sp_addserver	"Adding a remote server" on page 481
Manage remote server names	sp_addserver	"Managing remote server names" on page 482
Change server connection options	sp_serveroption	"Setting server connection options" on page 483
Display information about servers	sp_helpserver	"Getting information about servers" on page 485
Drop a server	sp_dropserver	"Dropping remote servers" on page 485

Adding a remote server

A System Security Officer uses sp_addserver to add entries to the sysservers table. On the server originating the call, you must add one entry for the local server, and one for each remote server that your server will call.

When you create entries for a remote server, you can either:

- Refer to them by the name listed in the interfaces file, or
- Provide a local name for the remote server. For example, if the name in the interfaces file is "MAIN_PRODUCTION," you may want to call it simply "main."

The syntax is:

```
sp_addserver lname [{, local | null}
    [, pname]]
```

where:

- *lname* provides the local "call name" for the remote server. If this name is *not* the same as the remote server's name in the interfaces file, you must provide that name as the third parameter, *pname*.

  The remote server must be listed in the interfaces file on the local machine. If it's not listed, copy the interfaces file entry from the remote server and append it to your existing interfaces file. Be sure to keep the same port numbers.

- local identifies the server being added as a local server. The local value is used only after start-up, or after a reboot, to identify the local server name so that it can appear in messages printed out by Adaptive Server. null specifies that this server is a remote server.

  Note For users to be able to run RPCs successfully from the local server, the local server must be added with the local option and rebooted. The rebooting is required to set the global variable @@servername.

- *pname* is the remote server listed in the interfaces file for the server named *lname*. This optional argument permits you to establish local aliases for any other Adaptive Server, Open Server™, or Backup Server that you may need to communicate with. If you do not specify *pname*, it defaults to *lname*.

Examples of adding remote servers

This example creates an entry for the local server named DOCS:

```
sp_addserver DOCS, local
```

The next example creates an entry for a remote server named GATEWAY:

```
sp_addserver GATEWAY
```

To run a remote procedure such as sp_who on the GATEWAY server, execute either:

```
GATEWAY.sybsytemprocs.dbo.sp_who
```

or:

```
GATEWAY...sp_who
```

This example gives a remote server called MAIN_PRODUCTION the local alias “main:”

```
sp_addserver main, null, MAIN_PRODUCTION
```

The user can then enter:

```
main...sp_who
```

Managing remote server names

The master.dbo.sysservers table has two name columns:

- srvname is the unique server name that users must supply when executing remote procedure calls.
- srvnetname is the server's network name, which must match the name in the *interfaces* file.

To add or drop servers from your network, you can use sp_addserver to update the server's network name in srvnetname.

For example, to remove the server MAIN from the network, and move your remote applications to TEMP, you can use the following statement to change the network name, while keeping the local alias:

```
sp_addserver MAIN, null, TEMP
```

sp_addserver displays a message telling you that it is changing the network name of an existing server entry.

Setting server connection options

sp_serveroption sets the server options timeouts, net password encryption, rpc security model A, and rpc security model B, which affect connections with remote servers. Additionally, if you have set the remote procedure security model to rpc security model B, you can use sp_serveroption to set these additional options: security mechanism, mutual authentication, use message confidentiality, and use message integrity.

The options you specify for sp_serveroption do not affect the communication between Adaptive Server and Backup Server.

The following sections describe timeouts, net password encryption, rpc security model A, and rpc security model B. For information about the additional options you can specify when rpc security model B is on, see "Establishing security for remote procedures" on page 514.

Using the *timeouts* option

A System Administrator can use the timeouts option to disable and enable the normal timeout code used by the local server.

By default, timeouts is set to true, and the site handler process that manages remote logins times out if there has been no remote user activity for one minute. By setting timeouts to false on both of the servers involved in remote procedure calls, the automatic timeout is disabled. This example changes timeouts to false:

```
sp_serveroption GATEWAY, "timeouts", false
```

After you set timeouts to false on both servers, when a user executes an RPC in either direction, the site handler on each machine runs until one of the servers is shut down. When the server is brought up again, the option remains false, and the site handler will be reestablished the next time a user executes an RPC. If users execute RPCs frequently, it is probably efficient in terms of system resources to set this option to false, since there is some system overhead involved in setting up the physical connection.

Using the *net password encryption* option

A System Security Officer can use net password encryption to specify whether connections with a remote server are to be initiated with a client-side password encryption handshake or with the usual unencrypted password handshake sequence. The default is false.

If net password encryption is set to true:

1 The initial login packet is sent without passwords.

2 The client indicates to the remote server that encryption is desired.

3 The remote server sends back an encryption key, which the client uses to encrypt its plain text passwords.

4 The client then encrypts its own passwords, and the remote server uses the key to authenticate them when they arrive.

This example sets net password encription to true:

```
sp_serveroption GATEWAY, "net password encryption",
    true
```

This option does not affect Adaptive Server's interaction with Backup Server.

Using the *rpc security model* options

The rpc security model A and rpc security model B options determine what kind of security is available for RPCs. If you use model A, which is the default, Adaptive Server does not support security services such as message confidentiality via encryption between the two servers.

For security model B, the local Adaptive Server gets a credential from the security mechanism and uses the credential to establish a secure physical connection with the remote Adaptive Server. With this model, you can choose one or more of these security services: mutual authentication, message confidentiality via encryption, and message integrity.

To set security model A for the server GATEWAY, execute:

```
sp_serveroption GATEWAY, "rpc security model A",
   true
```

For information about how to set up servers for Security Model B, see "Establishing security for remote procedures" on page 514.

Getting information about servers

sp_helpserver reports on servers. Without an argument, it provides information about all the servers listed in sysservers. When you include a server name, it provides information about that server only. The syntax is:

sp_helpserver [*server*]

sp_helpserver checks for both srvname and srvnetname in the master..sysremotelogins table.

For operating-system-specific information about setting up remote servers, see the the installation documentation for your platform.

Dropping remote servers

A System Security Officer can use the sp_dropserver system procedure to drop servers from sysservers. The syntax is:

sp_dropserver *server* [, droplogins]

where:

- *server* is the name of the server you want to drop.
- droplogins allows you to drop a remote server and all of that server's remote login information in one step. If you do not use droplogins, you cannot drop a server that has remote logins associated with it.

The following statement drops the GATEWAY server and all of the remote logins associated with it:

```
sp_dropserver GATEWAY, droplogins
```

You don't have to use droplogins if you want to drop the local server; that entry does not have remote login information associated with it.

Adding remote logins

The System Security Officer and System Administrator of any Adaptive Server share control over which remote users can access the server, and what identity the remote users assume. The System Administrator uses sp_addremotelogin to add remote logins and sp_dropremotelogin to drop remote logins. The System Security Officer uses sp_remoteoption to control whether password checking will be required.

Mapping users' server IDs

Logins from a remote server can be mapped to a local server in three ways:

- A particular remote login can be mapped to a particular local login name. For example, user "joe" on the remote server might be mapped to "joesmith".
- All logins from one remote server can be mapped to one local name. For example, all users sending remote procedure calls from the MAIN server might be mapped to "remusers".
- All logins from one remote server can use their remote names.

The first option can be combined with the other two options, and its specific mapping takes precedence over the other two more general mappings. The second and third options are mutually exclusive; you can use either of them, but not both.

To change the mapping option:

Use sp_dropremotelogin to remove the old mapping.

Use sp_addremotelogin to add remote logins. The syntax is:

```
sp_addremotelogin remoteserver [, loginame
    [, remotename]]
```

If the local names are not listed in master..syslogins, add them as Adaptive Server logins with sp_addlogin before adding the remote logins.

Only a System Administrator can execute sp_addremotelogin. For more information, see the *Adaptive Server Reference Manual*.

Mapping remote logins to particular local names

The following example maps the login named “pogo” from a remote system to the local login name “bob”. The user logs in to the remote system as “pogo”. When that user executes remote procedure calls from GATEWAY, the local system maps the remote login name to “bob”.

```
sp_addlogin bob
sp_addremotelogin GATEWAY, bob, pogo
```

Mapping all remote logins to one local name

The following example creates an entry that maps all remote login names to the local name “albert”. All names are mapped to “albert”, except those with specific mappings, as described in the previous section. For example, if you mapped “pogo” to “bob”, and then the rest of the logins to “albert”, “pogo” still maps to “bob”.

```
sp_addlogin albert
sp_addremotelogin GATEWAY, albert
```

If you use sp_addremotelogin to map all users from a remote server to the same local name, use sp_remoteoption to specify the “trusted” option for those users. For example, if all users from server GATEWAY that are mapped to “albert” are to be trusted, specify:

```
sp_remoteoption GATEWAY, albert, NULL, trusted, true
```

If you do not specify the logins as trusted, the logins will not be allowed to execute RPCs on the local server unless they specify passwords for the local server when they log in to the remote server. Users, when they use Open Client Client-Library can use the routine ct_remote_pwd to specify a password for server-to-server connections. isql and bcp do not permit users to specify a password for RPC connections. See “Password checking for remote users” on page 490 for more information about sp_remoteoption.

Warning! Do not map more than one remote login to a single local login, as it reduces individual accountability on the server. Audited actions can be traced only to the local server login, not to the individual logins on the remote server.

If you are using network based security

If users are logged into the remote server using “unified login”, the logins must also be trusted on the local server, or they must specify passwords for the server when they log into the remote server. For information about “unified login”, see “Using unified login” on page 507

Warning! Using the trusted mode of sp_remoteoption reduces the security of your server, as passwords from such “trusted” users are not verified.

Keeping remote login names for local servers

To enable remote users to keep their remote login names while using a local server:

1 Use sp_addlogin to create a login for each login from the remote server.

2 Use sp_addremotelogin for the server as a whole to create an entry in master..sysremotelogins with a null value for the remote login name and a value of -1 for the suid. For example:

```
sp_addremotelogin GATEWAY
```

Example of remote user login mapping

This statement displays the local and remote server information recorded in master..sysservers:

```
select srvid, srvname from sysservers
srvid  srvname
-----  ----------
    0  SALES
    1  CORPORATE
    2  MARKETING
    3  PUBLICATIONS
    4  ENGINEERING
```

The SALES server is local. The other servers are remote.

This statement displays information about the remote servers and users stored in master..sysremotelogins:

```
select remoteserverid, remoteusername, suid
  from sysremotelogins
```

```
remoteserverid   remoteusername   suid
--------------   --------------   ------
1                joe              1
1                nancy            2
1                NULL             3
3                NULL             4
4                NULL             -1
```

By matching the value of remoteserverid in this result and the value of srvid in the previous result, you can find the name of the server for which the remoteusername is valid. For example, in the first result, srvid 1 indicates the CORPORATE server; in the second result remoteserverid 1 indicates that same server. Therefore, the remote user login names "joe" and "nancy" are valid on the CORPORATE server.

The following statement shows the entries in master..syslogins:

```
select suid, name from syslogins
suid    name
------  ------------
    1   sa
    2   vp
    3   admin
    4   writer
```

The results of all three queries together show:

- The remote user name "joe" (suid 1) on the remote CORPORATE server (srvid and remoteserverid 1) is mapped to the "sa" login (suid 1).
- The remote user name "nancy" (suid 2) on the remote CORPORATE server (srvid and remoteserverid 1) is mapped to the "vp" login (suid 2).
- The other logins from the CORPORATE server (remoteusername "NULL") are mapped to the "admin" login (suid 3).
- All logins from the PUBLICATIONS server (srvid and remoteserverid 3) are mapped to the "writer" login (suid 4).
- All logins from the ENGINEERING server (srvid and remoteserverid 4) are looked up in master..syslogins by their remote user names (suid -1).
- There is no remoteserverid entry for the MARKETING server in sysremotelogins. Therefore, users who log in to the MARKETING server cannot run remote procedure calls from that server.

The remote user mapping procedures and the ability to set permissions for individual stored procedures give you control over which remote users can access local procedures. For example, you can allow the "vp" login from the CORPORATE server to execute certain local procedures and all other logins from CORPORATE to execute the procedures for which the "admin" login has permission.

Note In many cases, the passwords for users on the remote server must match passwords on the local server.

Password checking for remote users

A System Security Officer can use sp_remoteoption to determine whether passwords will be checked when remote users log in to the local server. By default, passwords are verified ("untrusted" mode). In trusted mode, the local server accepts remote logins from other servers and front-end applications without user-access verification for the particular login.

When sp_remoteoption is used with arguments, it changes the mode for the named user. The syntax is:

```
sp_remoteoption [remoteserver, loginame, remotename,
    optname, {true | false}]
```

The following example sets trusted mode for the user "bob":

```
sp_remoteoption GATEWAY, pogo, bob, trusted,
    true
```

Effects of using the untrusted mode

The effects of the "untrusted" mode depend on the user's client program. isql and some user applications require that logins have the same password on the remote server and the local server. Open Client™ applications can be written to allow local logins to have different passwords on different servers.

To change your password in "untrusted" mode, you must first change it on all the remote systems you access before changing it on your local server. This is because of the password checking. If you change your password on the local server first, when you issue the remote procedure call to execute sp_password on the remote server your passwords will no longer match.

The syntax for changing your password on the remote server is:

remote_server...sp_password *caller_passwd*, *new_passwd*

On the local server, the syntax is:

sp_password *caller_passwd*, *new_passwd*

See "Changing passwords" on page 369 for more information about changing your password.

Getting information about remote logins

sp_helpremotelogin prints information about the remote logins on a server. The following example shows the remote login "pogo" mapped locally to login name "bob", with all other remote logins keeping their remote names.

```
sp_helpremotelogin
server     remote_user_name     local_user_name      options
-------    ----------------     -----------------    --------
GATEWAY    **mapped locally**   **use local name**   untrusted
GATEWAY    pogo                 bob                  untrusted
```

Configuration parameters for remote logins

Table 13-2 shows the configuration parameters that affect RPCs. All these configuration parameters are set using sp_configure and do not take effect until Adaptive Server is restarted.

Table 13-2: Configuration parameters that affect RPCs

Configuration Parameter	Default
allow remote access	1
number of remote logins	20
number of remote sites	10
number of remote connections	20
remote server pre-read packets	3

Allowing remote access

To allow remote access to or from a server, including Backup Server, set allow remote access to 1:

```
sp_configure "allow remote access", 1
```

To disallow remote access at any time, set allow remote access to 0:

```
sp_configure "allow remote access", 0
```

Only a System Security Officer can set the allow remote access parameter.

Note You cannot perform database or transaction log dumps while the allow remote access parameter is set to 0.

Controlling the number of active user connections

To set the number of active user connections from this site to remote servers, use number of remote logins. This command sets number of remote logins to 50:

```
sp_configure "number of remote logins", 50
```

Only a System Administrator can set the number of remote logins parameter.

Controlling the number of remote sites

To control the number of remote sites that can access a server simultaneously, use number of remote sites. All accesses from an individual site are managed by one site handler. This parameter controls the number of site handlers, not the number of individual, simultaneous procedure calls. For example, if you set number of remote sites to 5, and each site initiates three remote procedure calls, sp_who shows 5 site handler processes for the 15 processes. Only a System Administrator can set the number of remote sites.

Controlling the number of active remote connections

To control the limit on active remote connections that are initiated to and from a server, use the number of remote connections parameter. This parameter controls connections initiated from the server and connections initiated from remote sites to the server. Only a System Administrator can set number of remote connections.

Controlling number of preread packets

To reduce the needed number of connections, all communication between two servers is handled through one site handler. This site handler can preread and keep track of data packets for each user before the user process that needs them is ready.

To control how many packets a site handler will preread, use remote server pre-read packets. The default value, 3, is adequate in all cases; higher values can use too much memory. Only a System Administrator can set remote server pre-read packets. For more information, see “remote server pre-read packets” on page 157.

CHAPTER 14
Using Kerberos, DCE, and Windows NT LAN Manager

This chapter describes the network-based security services that enable you to authenticate users and protect data transmitted among machines on a network.

For information about the Secure Socket Layer (SSL) security mechanism, see Chapter 9, "Security Administration."

Topics covered in this chapter include:

Overview

In a distributed client/server computing environment intruders can view or tamper with confidential data. Adaptive Server works with third-party providers to give you security services that:

- Authenticate users, clients, and servers – Make sure they are who they say they are.
- Provide data confidentiality with encryption – Ensure that data cannot be read by an intruder.

- Provide data integrity – Prevent data tampering and detect when it has occurred

Table 14-1 lists the security mechanisms supported by Adaptive Server on UNIX and desktop platforms:

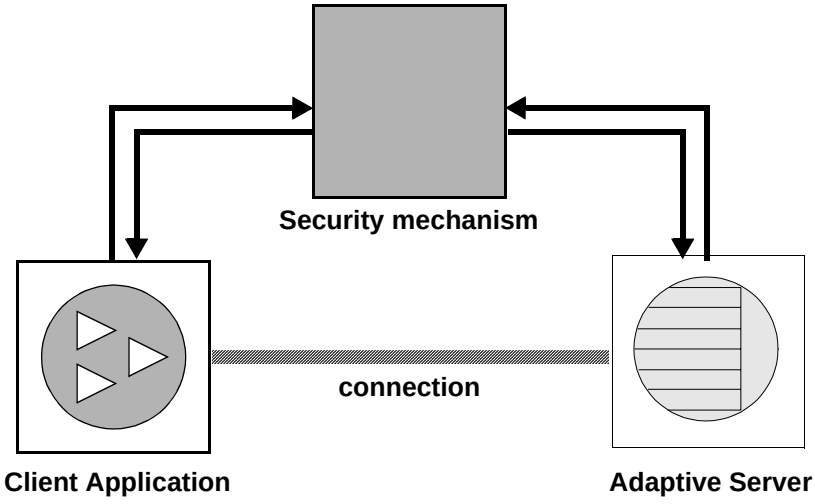
Table 14-1: Security mechanisms supported by Adaptive Server

UNIX platforms	Desktop platforms
Distributed Computing Environment (DCE)	Windows NT LAN Manager
CyberSAFE Kerberos	

How applications use security services

The following illustration shows a client application using a security mechanism to ensure a secure connection with Adaptive Server.

Figure 14-1: Establishing secure connections between a client and Adaptive Server



The secure connection between a client and a server can be used for:

- Login authentication

- Message protection

Login authentication

If a client requests authentication services:

1 The client validates the login with the security mechanism. The security mechanism returns a *credential*, which contains security-relevant information.

2 The client sends the credential to Adaptive Server.

3 Adaptive Server authenticates the client's credential with the security mechanism. If the credential is valid, a secure connection is established between the client and Adaptive Server.

Message protection

If the client requests message protection services:

1 The client uses the security mechanism to prepare the data packet it will send to Adaptive Server.

   Depending upon which security services are requested, the security mechanism might encrypt the data or create a cryptographic signature associated with the data.

2 The client sends the data packet to Adaptive Server.

3 When Adaptive Server receives the data packet, it uses the security mechanism to perform any required decryption and validation.

4 Adaptive Server returns results to the client, using the security mechanism to perform the security functions that were requested; for example, Adaptive Server may return the results in encrypted form.

Security services and Adaptive Server

Depending upon the security mechanism you choose, Adaptive Server allows you to use one or more of these security services:

- Unified login –Authenticate users *once* without requiring them to supply a name and password every time they log in to an Adaptive Server.
- Message confidentiality – Encrypt data over the network.

- Mutual authentication – Verify the identity of the client and the server. This must be requested by the client and cannot be required by Adaptive Server.
- Message integrity – Verify that data communications have not been modified.
- Replay detection – Verify that data has not been intercepted by an intruder.
- Out-of-sequence check – Verify the order of data communications.
- Message origin checks – Verify the origin of the message.
- Remote procedure security – Establish mutual authentication, message confidentiality, and message integrity for remote procedure communications.

Note The security mechanism you are using may not all of these services. For information about what services are available to you, see "Getting information about available security services" on page 526.

Administering network-based security

Table 14-2 provides an overall process for using the network-based security functions provided by Adaptive Server. You must install Adaptive Server before you can complete the steps in Table 14-2.

Table 14-2: Process for administering network-based security

Step	Description	See
1. Set up the configuration files: • *libtcl.cfg* • *objectid.dat* • *interfaces* (or Directory Service)	Edit the *libtcl.cfg* file. Edit the *objectid.dat* file. Edit the *interfaces* file or Directory Service.	• "Setting up configuration files for security" on page 499 • The *Open Client/Server Configuration Guide* for your platform.
2. Make sure the security administrator for the security mechanism has created logins for each user and for the Adaptive Server and Backup Server.	The security administrator must add names and passwords for users and servers in the security mechanism. For DCE, the security administrator needs to create a *keytab* file for server entries.	• The documentation supplied with your security mechanism. • "Identifying users and servers to the security mechanism" on page 506.

Step	Description	See
3. Configure security for your installation.	Use sp_configure.	"Configuring Adaptive Server for security" on page 506.
4. Restart Adaptive Server.	Activates the use security services parameter.	"Restarting the server to activate security services" on page 512.
5. Add logins to Adaptive Server to support enterprise-wide login.	Use sp_addlogin to add users. Optionally, specify a default secure login with sp_configure.	"Adding logins to support unified login" on page 513.
6. Determine the security model for remote procedures and set up the local and remote servers for RPC security.	Use sp_serveroption to choose the security model (A or B).	"Establishing security for remote procedures" on page 514.
7. Connect to the server and use security services.	Use isql_dce or isql_r (if you are using DCE library services or security services) or Open Client Client-Library to connect to Adaptive Server, specifying the security services you want to use.	• "Connecting to the server and using the security services" on page 523 • The *Open Client/Server Configuration Guide* for your platform. • "Security Features" topics page in the *Open Client Client-Library/C Reference Manual.*
8. Check the security services and security mechanisms that are available.	Use the functions show_sec_services and is_sec_services_on to check which security services are available. For a list of security mechanisms and their security services supported by Adaptive Server, use select to query the syssecmechs system table.	"Getting information about available security services" on page 526.

Setting up configuration files for security

Configuration files are created during installation at a default location in the Sybase directory structure. Table 14-3 provides an overview of the configuration files required for using network-based security.

Table 14-3: Names and locations for configuration files

File name	Description	Location
libtcl.cfg	The driver configuration file contains information regarding directory, security, and network drivers and any required initialization information.	UNIX platforms: *$SYBASE/config* Desktop platforms: *SYBASE_home\ini*
objectid.dat	The object identifiers file maps global object identifiers to local names for character set, collating sequence, and security mechanisms.	UNIX platforms: *$SYBASE/config* Desktop platforms: *SYBASE_home\ini*
UNIX: *interfaces* Desktop Platforms: sql.ini	The interfaces file contains connection and security information for each server listed in the file. **Note** In this release, you can use a Directory Service instead of the interfaces file.	UNIX platforms: *$SYBASE* Desktop platforms: *SYBASE_home\ini*

For a detailed description of the configuration files, see the *Open Client/Server Configuration Guide* for your platform.

Preparing*libtcl.cfg*to use network-based security

libtcl.cfg contains information about three types of drivers:

- Network (Net-Library)
- Directory Services
- Security

A **driver** is a Sybase library that provides an interface to an external service provider. Drivers are dynamically loaded so that you can change the driver used by an application without re-linking the application.

Entries for network drivers

The syntax for a network driver entry is:

driver=protocol description

where:

- *driver* is the name of the network driver.
- *protocol* is the name of the network protocol.
- *description* is a description of the entry. This element is optional.

Note If you do not specify a network driver, an appropriate driver for your application and platform is automatically used. For example, for UNIX platforms, a driver that can handle threads is automatically chosen when security services are being used.

Entries for Directory Services

Entries for Directory Services apply if you want to use a Directory Service instead of the interfaces file. For information about directory entries, see the configuration documentation for your platform, and the *Open Client/Server Configuration Guide* for your platform.

Entries for security drivers

The syntax for a security driver entry is:

```
provider=driver init-string
```

where:

- *provider* is the local name for the security mechanism. The mapping of the local name to a global object identifier is defined in *objectid.dat*.

  The default local names are:

  - "dce" – For the DCE security mechanism.
  - "csfkrb5" – For the CyberSAFE Kerberos security mechanism.
  - "LIBSMSSP" – For Windows LAN Manager on Windows NT or Windows 95 (clients only).

  If you use a local mechanism name other than the default, you must change the local name in the objectid.dat file (see "The objectid.dat file" on page 504 for an example).

- *driver* is the name of the security driver. The default location of all drivers for UNIX platforms is *$SYBASE/lib*. The default location for desktop platforms is SYBASE_home\dll.

- *init-string* is an initialization string for the driver. This element is optional. The value for *init-string* varies by driver:
  - For the DCE driver, the syntax for *init-string* is:

    secbase=/.../*cell_name*

    where *cell_name* is the name of your DCE cell.
  - For the CyberSAFE Kerberos driver, the syntax for *init-string* is:

    secbase=@*realm*

    where *realm* is the default CyberSAFE Kerberos realm name.
  - For the Windows NT LAN Manager, *init-string* is not applicable.

UNIX platform information

This section contains information specific to UNIX platforms. For more information, see the *Open Client/Server Configuration Guide for UNIX*.

For UNIX platforms, no special tools for editing the *libtcl.cfg* file are available. Use your favorite editor to comment and uncomment the entries that are already in place after you install Adaptive Server.

The *libtcl.cfg* file, after installation of Adaptive Server on a UNIX platform, already contains entries for the three sections of the file:

- [DRIVERS]
- [DIRECTORY]
- [SECURITY]

The sections do not have to be in a specific order.

Make sure that the entries you do not want to use are commented (begin with ";") and the entries you want are uncommented (do not begin with ";").

Sample *libtcl.cfg* File for Sun Solaris

```
[DRIVERS]
;libtli.so=tcp unused ; This is the non-threaded tli driver.
;libtli_r.so=tcp unused ; This is the threaded tli driver.


[DIRECTORY]
;dce=libddce.so ditbase=/.:/subsys/sybase/dataservers
;dce=libddce.so ditbase=/.:/users/cfrank
```

```
[SECURITY]
dce=libsdce.so secbase=/.../svrsole4_cell
```

This *libtcl.cfg* file is set up to use the DCE security service. Notice that this file does not use Directory Services because all [DIRECTORY] section entries are commented.

Because all entries in the [DRIVERS] section for network drivers are also commented, appropriate drivers are chosen automatically by the system. A threaded driver is chosen automatically when security services are being used, and a non-threaded driver is chosen automatically for applications that cannot work with threaded drivers. For example, Backup Server does not support security services and does not work with a threaded driver.

Desktop platform information

This section contains information specific to desktop platforms. For more information, see the *Open Client/Server Configuration Guide for Desktop Platforms*.

Use the ocscfg utility to edit the *libtcl.cfg* file. See the *Open Client/Server Configuration Guide for Desktop Platforms* for instructions for using ocscfg.

The ocscfg utility creates section headings automatically for the *libtcl.cfg* file.

Sample *libtcl.cfg* file for desktop platforms

```
[NT_DIRECTORY]
ntreg_dsa=LIBDREG
ditbase=software\sybase\serverdsa

[DRIVERS]
NLWNSCK=TCP  Winsock TCP/IP Net-Lib driver
NLMSNMP=NAMEPIPE  Named Pipe Net-Lib driver
NLNWLINK=SPX  NT NWLINK SPX/IPX Net-Lib driver
NLDECNET=DECNET  DecNET Net-Lib driver

[SECURITY]
NTLM=LIBSMSSP
```

The *objectid.dat* file

The objectid.dat file maps global object identifiers, such as the one for the DCE service (“1.3.6.1.4.1.897.4.6.1”) to local names, such as “dce”. The file contains sections such as [CHARSET] for character sets and [SECURITY] for security services. Of interest here is the security section. Following is a sample objectid.dat file:

```
[secmech]
        1.3.6.1.4.1.897.4.6.1   = dce
        1.3.6.1.4.1.897.4.6.3   = NTLM
        1.3.6.1.4.1.897.4.6.6   = csfkrb5
```

You need to change this file only if you have changed the local name of a security service in the *libtcl.cfg* file. Use a text editor to edit the file.

For example, if you changed

```
[SECURITY]
dce=libsdce.so secbase=/.../svrsole4_cell
```

to

```
[SECURITY]
dce_group=libsdce.so secbase=/.../svrsole4_cell
```

in *libtcl.cfg*, then you need to change the objectid.dat file to reflect the change. Simply change the local name in the line for DCE in objectid.dat:

```
1.3.6.1.4.1.897.4.6.1   = dce_group
```

Note You can specify only one local name per security mechanism.

Specifying security information for the server

You can choose to use an *interfaces* file or a *Directory Service* to provide information about the servers in your installation.

The interfaces file contains network and security information for servers. If you plan to use security services, the interfaces file must include a “secmech” line, which gives the global identifier or identifiers of the security services you plan to use.

Instead of using the interfaces file, Adaptive Server supports Directory Services to keep track of information about servers. A Directory Service manages the creation, modification, and retrieval of information about network servers. The advantage of using a Directory Service is that you do not need to update multiple interfaces files when a new server is added to your network or when a server moves to a new address. If you plan to use security services with a Directory Service, the secmech security attribute must be defined. It must point to one or more global identifiers of the security services you plan to use.

UNIX tools for specifying the security mechanism

To specify the security mechanism or mechanisms you want to use:

- If you are using the interfaces file, use the dscp utility.
- If you are using a Directory Service, use the dscp_r or dscp_dce utility.

Note The dsedit tool, which helps you create entries for either the interfaces file or a Directory Service, is available on UNIX platforms. However, it does not support the creation of secmech entries for security mechanisms.

For more information about dscp, see the *Open Client/Server Configuration Guide for UNIX*.

Desktop tools for specifying server attributes

To provide information about the servers for your installation in the sql.ini file or a Directory Service, use the dsedit utility. This utility provides a graphical user interface for specifying server attributes such as the server version, name, and security mechanism. For the security mechanism attribute, you can specify one or more object identifiers for the security mechanisms you plan to use. For information about using dsedit, see the *Open Client/Server Configuration Guide for Desktop Platforms*.

Identifying users and servers to the security mechanism

The security administrator for the security mechanism must define *principals*, which include both users and servers, to the security mechanism. Table 14-4 lists tools you can use to add users and servers.

Table 14-4: Defining users and servers to the security mechanism

Security mechanism	Command or tool
DCE	Use the DCE dcecp tool's user create command to create a new principal (user or server). In addition, use the keytab create command to create a DCE keytab file, which contains a principal's password in encrypted form. When you are defining a server to DCE, use command options that specify that the new principal can act as a server.
CyberSAFE Kerberos	Use the CyberSAFE kadmin utility's add command. In addition, use the kadmin utility, with the ext command to create a key in a CyberSAFE Kerberos server key table file. When you are defining a server to CyberSAFE Kerberos, use command options that specify that the new principal can act as a server.
Windows NT LAN Manager	Run the User Manager tool to define users to the Windows NT LAN Manager. Be sure to define the Adaptive Server name as a user to Windows NT LAN Manager and bring up Adaptive Server as that user name.

Note In a production environment, you must control the access to files that contain the keys of the servers and users. If users can access the keys, they can create a server that impersonates your server.

Refer to the documentation available from the third-party provider of the security mechanism for detailed information about how to perform required administrative tasks.

Configuring Adaptive Server for security

Adaptive Server includes several configuration parameters for administering network-based security. To set these parameters, you must be a System Security Officer. All parameters for network-based security are part of the "Security-Related" configuration parameter group.

Configuration parameters are used to:

- Enable network-based security
- Require unified login
- Require message confidentiality with data encryption
- Require one or more message integrity security services

Enabling network-based security

To enable or disable network-based security, use sp_configure to set the use security services configuration parameter. Set this parameter to 1 to enable network-based security. If this parameter is 0 (the default), network-based security services are not available. The syntax is:

```
sp_configure "use security services", [0|1]
```

For example, to enable security services, execute:

```
sp_configure “use security services”, 1
```

Note This configuration parameter is static; you must restart Adaptive Server for it to take effect. See “Restarting the server to activate security services” on page 512.

Using unified login

Configuration parameters are available to:

- Require unified login
- Establish a default secure login

All the parameters for unified login take effect immediately. You must be a System Security Officer to set the parameters.

Requiring unified login

To require all users to already be authenticated by a security mechanism, set the unified login required configuration parameter to 1. If this parameter is 0 (the default), Adaptive Server will accept traditional login names and passwords, as well as already-authenticated credentials. The syntax is:

```
sp_configure "unified login required", [0|1]
```

For example, to require all logins to be authenticated by a security mechanism, execute:

```
sp_configure “unified login required”, 1
```

Establishing a secure default login

When a user with a valid credential from a security mechanism logs in to Adaptive Server, the server checks whether the user name exists in master..syslogins. If it does, that user name is used by Adaptive Server. For example, if a user logs in to the DCE security mechanism as “ralph,” and “ralph” is a name in master..syslogins, Adaptive Server uses all roles and authorizations defined for “ralph” in the server.

However, if a user with a valid credential logs into Adaptive Server, but is unknown to the server, the login is accepted only if a *secure default login* is defined with sp_configure. Adaptive Server uses the default login for any user who is not defined in master..syslogins, but who is pre-authenticated by a security mechanism. The syntax is:

sp_configure "secure default login", 0, *login_name*

The default value for secure default login is “guest.”

This login must be a valid login in master..syslogins. For example, to set the login “gen_auth” to be the default login:

1 Use sp_addlogin to add the login as a valid user in Adaptive Server:

```
sp_addlogin gen_auth, pwgenau
```

This procedure sets the initial password to “pwgenau”

2 Use sp_configure to designate the login as the security default.

```
sp_configure "secure default login", 0, gen_auth
```

Adaptive Server will use this login for a user who is pre-authenticated by a security mechanism but is unknown to Adaptive Server.

Note More than one user can assume the suid associated with the secure default login. Therefore, you might want to activate auditing for all activities of the default login. You may also want to consider using sp_addlogin to add all users to the server.

For more information about adding logins, see “Adding logins to support unified login” on page 513 and “Adding logins to Adaptive Server” on page 345.

Mapping security mechanism login names to server names

Some security mechanisms may allow login names that are not valid in Adaptive Server. For example, login names that are longer than 30 characters, or login names containing special characters such as !, %, *, and & are invalid names in Adaptive Server. All login names in Adaptive Server must be valid identifiers. For information about what identifiers are valid, see Chapter 7, "Expressions, Identifiers, and Wildcard Characters," in the *Adaptive Server Reference Manual*.

Table 14-5 shows how Adaptive Server converts invalid characters in login names:

Table 14-5: Conversion of invalid characters in login names

Invalid characters	Converts to
Ampersand &	Underscore _
Apostrophe ’	
Backslash \	
Colon :	
Comma ,	
Equals sign =	
Left quote ‘	
Percent %	
Right angle bracket >	
Right quote ’	
Tilde ~	
Caret ^	Dollar sign $
Curly braces { }	
Exclamation point !	
Left angle bracket <	
Parenthesis ( )	
Period .	
Question mark ?	
Asterisk *	Pound sign #
Minus sign -	
Pipe |	
Plus sign +	
Quotation marks "	
Semicolon ;	
Slash /	
Square brackets [ ]	

Requiring message confidentiality with encryption

To require all messages into and out of Adaptive Server to be encrypted, set the msg confidentiality reqd configuration parameter to 1. If this parameter is 0 (the default), message confidentiality is not required but may be established by the client.

The syntax for setting this parameter is:

```
sp_configure configuration_parameter, [0 | 1]
```

For example, to require that all messages be encrypted, execute:

```
sp_configure "msg confidentiality reqd", 1
```

Requiring data integrity

Adaptive Server allows you to use the following configuration parameters to require that one or more types of data integrity be checked for all messages:

- msg integrity reqd – set this parameter to 1 to require that all messages be checked for general tampering. If this parameter is 0 (the default), message integrity is not required but may be established by the client if the security mechanism supports it.

Memory requirements for network-based security

Allocate approximately 2K additional memory per secure connection. The value of the max total_memory configuration parameter specifies the amount of memory that Adaptive Server requires at start-up. For example, if your server uses 2K logical pages, and if you expect the maximum number of secure connections occurring at the same time to be 150, increase the max total_memory parameter by 150, which increases memory allocation by 150 2K blocks.

The syntax is:

sp_configure “max total_memory”, *value*

For example, if Adaptive Server requires 75,000 2K blocks of memory, including the increased memory for network-based security, execute:

```
sp_configure "max total_memory", 75000
```

For information about estimating and specifying memory requirements, see the Chapter 18, “Configuring Memory.”

Restarting the server to activate security services

Once you have configured security services, you must restart Adaptive Server.

For Windows NT, see the configuration documentation for your platform.

For UNIX platforms, note that:

- After you complete the installation of Adaptive Server, your runserver file contains an invocation of the dataserver utility to start Adaptive Server.
- Two versions of the dataserver utility are available: dataserver_dce, and dataserver. Likewise, two versions of the diagserver are available: diagserver_dce and diagserver. The utility you use depends on the platform you use:
  - For Sun Solaris platforms, use dataserver if you plan to use security services and dataserver if you do not plan to use security services.
  - For HP and RS/6000 platforms, use dataserver and diagserver. You can use a single binary, whether or not you are using security services.
- If you are using the DCE security service, be sure you have defined the keytab file. You can specify the -K option to dataserver_dce to specify the location of the keytab file. If you do not specify a location, Adaptive Server assumes the file is located in $SYBASE/config/$DSLISTEN_key. Optionally, you can specify the location as follows:

  ```
  $SYBASE/bin/dataserver_dce -Stest4 -dd_master
     -K/opt/dcelocal/keys/test4_key
  ```

  This dataserver_dce command boots the server using the master device d_master and the keytab file stored in */opt/dcelocal/keys/test4_key*.

  If you are using the default location for keytab, and $DSLISTEN is set to the name of your server (test4), you can execute:

  ```
  $SYBASE/bin/dataserver_dce -dd_master
  ```

  Then, Adaptive Server looks for the keytab file in $SYBASE/config/test4_key.

For information about setting up your keytab file for DCE, refer to the DCE administrative documentation.

Determining security mechanisms to support

use security services is set to 0, Adaptive Server supports no security mechanisms.

If use security services is set to 1, Adaptive Server supports a security mechanism when both of the following circumstances are true:

- The security mechanism's global identifier is listed in the interfaces file or Directory Service.
- The global identifier is mapped in *objectid.dat* to a local name that is listed in *libtcl.cfg*.

For information about how Adaptive Server determines which security mechanism to use for a particular client, see "Using security mechanisms for the client" on page 525.

Adding logins to support unified login

When users log in to Adaptive Server with a pre-authenticated credential, Adaptive Server:

1 Checks whether the user is a valid user in master..syslogins. If the user is listed in master..syslogins, Adaptive Server accepts the login without requiring a password.

2 If the user name is not in master..syslogins, Adaptive Server checks whether a default secure login is defined. If the default login is defined, the user is logged in successfully as that login. If a default login is not defined, Adaptive Server rejects the login.

Therefore, consider whether you want to allow only those users who are defined as valid logins to use Adaptive Server, or whether you want users to be able to login with the default login. You must add the default login in master..syslogins and use sp_configure to define the default. For details, see "Establishing a secure default login" on page 508.

General procedure for adding logins

Follow the general procedure described in Table 14-6 to add logins to the server and, optionally, to add users to one or more databases with appropriate roles and authorizations to one or more databases.

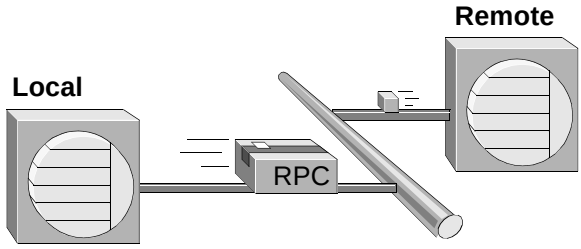
Table 14-6: Adding logins and authorizing database access

Task	Required role	Command or procedure	See
1. Add a login for the user.	System Security Officer	sp_addlogin	"Adding logins to Adaptive Server" on page 345
2. Add the user to one or more databases.	System Administrator or Database Owner	sp_adduser Execute this procedure from within the database.	"Adding users to databases" on page 348
3. Add the user to a group in a database.	System Administrator or Database Owner	sp_changegroup Execute this procedure from within the database.	• "Changing a user's group membership" on page 371 • sp_changegroup in the *Adaptive Server Reference Manual*
4. Grant system roles to the user.	System Administrator or System Security Officer	grant role	• "Creating and assigning roles to users" on page 355 • grant in the *Adaptive Server Reference Manual*
5. Create user-defined roles and grant the roles to users.	System Security Officer	create role grant role	• "Creating and assigning roles to users" on page 355 in the *Adaptive Server Reference Manual* • grant in the *Adaptive Server Reference Manual* • create role in the *Adaptive Server Reference Manual*
6. Grant access to database objects.	Database object owners		Chapter 11, "Managing User Permissions"

Establishing security for remote procedures

Adaptive Server acts as the client when it connects to another server to execute a remote procedure call (RPC) as shown in Figure 14-2.

Figure 14-2: Adaptive Server acting as client to execute an RPC



One *physical* connection is established between the two servers. The servers use the physical connection to establish one or more *logical* connections—one logical connection for each RPC.

Adaptive Server 11.5 supports two security models for RPCs: *security model A* and *security model B.*

Security model A

For security model A, Adaptive Server does not support security services such as message confidentiality via encryption between the two servers. Security Model A is the default.

Security model B

For security model B, the local Adaptive Server gets a credential from the security mechanism and uses the credential to establish a secure physical connection with the remote Adaptive Server. With this model, you can use one or more of these security services:

- Mutual authentication – the local server authenticates the remote server by retrieving the credential of the remote server and verifying it with the security mechanism. With this service, the credentials of both servers are authenticated and verified.
- Message confidentiality via encryption – messages are encrypted when sent to the remote server, and results from the remote server are encrypted.

- Message integrity – messages between the servers are checked for tampering.

Unified login and the remote procedure models

If the local server and remote server are set up to use security services, you can use unified login on both servers with *either* model, using one of these two methods:

- The System Security Officer defines a user as "trusted" with sp_remoteoption on the remote server. With this method, a security mechanism such as DCE authenticates the user and password. The user gains access to the local server via "unified login" and executes an RPC on the remote server. The user is trusted on the remote server and does not need to supply a password.
- A user specifies a password for the remote server when he or she connects to the local server. The facility to specify a remote server password is provided by the ct_remote_pwd routine available with Open Client Client-Library/C. For more information about this routine, see the *Open Client Client-Library/C Reference Manual*.

Establishing the security model for RPCs

To establish the security model for RPCs, use sp_serveroption. The syntax is:

sp_serveroption *server, optname,* [true | false]

To establish the security model, set *optname* to rpc security model A or rpc security model B. *server* names the remote server.

For example, to set model B for remote server TEST3, execute:

```
sp_serveroption test3, "rpc security model B", true
```

The default model is "A," that is, remote procedure calls are handled the same as in previous releases. No server options need to be set for model A.

Setting server options for RPC security model B

For RPC security model B, you can set options with the sp_serveroption system procedure. The syntax is:

sp_serveroption *server, optname,* optvalue

where:

- *server* is the name of the remote server.
- *optname* is the name of the option. Values can be:
  - security mechanism – the name of the security mechanism to use when running an RPC on a remote server.
  - mutual authentication – set this option to 1 to cause the local Adaptive Server to authenticate and verify the remote server. If this parameter is 0 (the default), the remote server still verifies the local server when it sends an RPC, but the local server does not check the validity of the remote server.
  - use message confidentiality – set this option to 1 to cause all messages for the RPCs to be encrypted when they are sent to the remote server and received from the remote server. If this parameter is 0 (the default), data for the RPCs will not be encrypted.
  - use message integrity – set this option to 1 to require that all RPC messages be checked for tampering. If this parameter is 0 (the default), RPC data will not be checked for tampering.
- *optvalue* must be equal to "true" or "false" for all values of *optname*, *except* security mechanism. If the option you are setting is security mechanism, specify the name of the security mechanism. To find the list of security mechanisms, execute:

  ```
  select * from syssecmechs
  ```

  For information about the syssecmechs system table, see "Determining enabled security services" on page 527.

For example, to set up the local server to execute RPCs on a remote server, TEST3, which will use the "dce" security mechanism, and to use mutual authentication for all RPCs between the two servers, execute:

```
sp_serveroption TEST3, "security mechanism", dce
sp_serveroption TEST3, "mutual authentication",
     true
```

Rules for setting up security model B for RPCs

Follow these rules when setting up security model B for RPCs:

- Both servers must be using security model B.
- Both servers must be using the same security mechanism, and that security mechanism must support the security services set with sp_serveroption.
- The System Security Officer of the local server must specify any security services that are required by the remote server. For example, if the remote server requires that all messages use the message confidentiality security service, the System Security Officer must use sp_serveroption to activate use message confidentiality.
- Logins who are authenticated by a security mechanism and log into Adaptive Server using "unified login" will not be permitted to execute RPCs on the remote procedure unless the logins are specified as "trusted" on the remote server or the login specifies the password for the remote server. Users, when they use Open Client Client-Library can use the routine ct_remote_pwd to specify a password for server-to-server connections. A System Administrator on Adaptive Server can use sp_remoteoption to specify that a user is trusted to use the remote server without specifying a password.

Preparing to use security model B for RPCs

Table 14-7 provides steps for using security model B to establish security for RPCS.

Table 14-7: Process for using security model B for RPCs

Task, who Performs it, and where	Command, system procedure, or tool	See
System Administrator from the operating system: 1. Make sure the interfaces file or the Directory Service contains an entry for both servers and a secmech line listing the security mechanism.	UNIX: dscp or dscp_dce Desktop: dsedit	"Specifying security information for the server" on page 504 For information about how to use dscp or dscp_dce, see *Open Client/Server Configuration Guide for UNIX*. For information about how to use dsedit, see the *Open Client/Server Configuration Guide for Desktop Platforms*.
System Security Officer on remote server: 2. Add the local server to master..sysservers.	sp_addserver Example: `sp_addserver "lcl_server"`	"Adding a remote server" on page 481. sp_addserver in the *Adaptive Server Reference Manual*.
System Security Officer on remote server: 3. Add logins to master..syslogins.	sp_addlogin Example: `sp_addlogin user1, "pwuser1"`	"Adding logins to Adaptive Server" on page 345. sp_addlogin in the *Adaptive Server Reference Manual*.

System Security Officer on remote server:
4. Set use security services on, and set the rpc security model B as the model for connections with the local server.
	sp_configure (to set use security services)
	sp_serveroption (to set the RPC security model)
	Example:
	`sp_configure "use security services", 1`
	`sp_serveroption lcl_server, "rpc security model B", true`
		"Establishing the security model for RPCs" on page 516.
		"Enabling network-based security" on page 507.
		use security services (Windows NT only) in Chapter 5, "Setting Configuration Parameters" in this manual.
		sp_configure in the *Adaptive Server Reference Manual*.
		sp_serveroption in the *Adaptive Server Reference Manual*.

Task, who Performs it, and where	Command, system procedure, or tool	See
System Administrator on remote server: 5. Optionally, specify certain users as "trusted" to log into the remote server from the local server without supplying a password.	sp_remoteoption Example: `sp_remoteoption lcl_server, user1, user1, trusted, true`	"Password checking for remote users" on page 490. sp_remoteoption in the *Adaptive Server Reference Manual*.

System Security Officer on local server:
6. Add both the local server and the remote server to master..sysservers.
	sp_addserver
	Example:
	`sp_addserver lcl_server, local`
	`sp_addserver rem_server`
		"Adding a remote server" on page 481.
		sp_addserver in the *Adaptive Server Reference Manual*.

System Security Officer on local server: 7. Add logins to master..logins.	sp_addlogin Example: `sp_addlogin user1, "pwuser1"`	"Adding logins to Adaptive Server" on page 345. sp_addlogin in the *Adaptive Server Reference Manual*.

System Security Officer on local server:
8. Set use security services on, and set the rpc security model B as the model for connections with the remote server.
	sp_configure (to set use security services)
	sp_serveroption (to set the RPC security model)
	Example:
	`sp_configure "use security services", 1`
	`sp_serveroption rem_server, "rpc security model B", true`
		"Establishing the security model for RPCs" on page 516.
		"Enabling network-based security" on page 507.
		use security services (Windows NT only) in Chapter 5, "Setting Configuration Parameters" in this manual.
		sp_configure in the *Adaptive Server Reference Manual*.
		sp_serveroption in the *Adaptive Server Reference Manual*.

System Security Officer on local server:
9. Specify the security mechanism and the security services to use for connections with the remote server.
	sp_serveroption
	Example:
	`sp_serveroption rem_server, "security mechanism", dce`
	`sp_serveroption rem_server, "use message integrity", true`
		"Setting server connection options" on page 483.
		sp_serveroption in the *Adaptive Server Reference Manual*.

Example of setting up security model B for RPCs

Assume that:

- A local server, lcl_serv, will run RPCs on a remote server, rem_serv.
- Both servers will use security model B and the DCE security service.
- These RPC security services will be in effect: mutual authentication and message integrity.
- Users “user1” and “user2” will use unified login to log in to the local server, lcl_serv, and run RPCs on rem_serv. These users will be “trusted” on rem_serv and will not need to specify a password for the remote server.
- User “user3” will not use unified login, will not be trusted, and must supply a password to Adaptive Server when logging in.

You would use the following sequence of commands to set up security for RPCs between the servers:

System Security Officer on remote server (rem_serv):

```
sp_addserver 'lcl_serv'
sp_addlogin user1, "eracg12"
sp_addlogin user2, "esirpret"
sp_addlogin user3, "drabmok"
sp_configure "use security services", 1
sp_serveroption lcl_serv, "rpc security model B",
   true
sp_serveroption lcl_serv, "security mechanism", dce
```

System Administrator on remote server (rem_serv):

```
sp_remoteoption lcl_serv, user1, user1, trusted,
   true
sp_remoteoption lcl_serv, user2, user2, trusted,
   true
```

System Security Officer on local server (lcl_serv):

```
sp_addserver lcl_serv, local
sp_addserver rem_serv
sp_addlogin user1, "eracg12"
sp_addlogin user2, "esirpret"
sp_addlogin user3, "drabmo1"
sp_configure "use security services", 1
sp_configure rem_serv, "rpc security model B", true
sp_serveroption rem_serv, "security mechanism", dce
sp_serveroption rem_serv, "mutual authentication"
```

```
      true
   sp_serveroption rem_serv, "use message integrity"
      true
```

In addition, the interfaces file or Directory Service must have entries for rem_serv and lcl_serv. Each entry should specify the "dce" security service. For example, you might have these interfaces entries, as created by the dscp utility:

```
## lcl_serv (3201)
lcl_serv
master tli tcp /dev/tcp \x00020c8182d6551100000000000000000
query tli tcp /dev/tcp \x00020c8182d6551100000000000000000
secmech 1.3.6.1.4.1.897.4.6.1
## rem_serv (3519)
rem_serv
master tli tcp /dev/tcp \x000214ad82d6551100000000000000000
query tli tcp /dev/tcp \x000214ad82d6551100000000000000000
secmech 1.3.6.1.4.1.897.4.6.1
```

Note To actually use the security services on either server, you must reboot the server so that the static parameter, use security services, takes effect.

For detailed information about setting up servers for remote procedure calls, see Chapter 13, "Managing Remote Servers."

Getting information about remote servers

The system procedure sp_helpserver displays information about servers. When it is used without an argument, it provides information about all the servers listed in sysservers. You can specify a particular server to receive information about that server. The syntax is:

sp_helpserver [*server*]

For example, to display information about the GATEWAY server, execute:

```
sp_helpserver GATEWAY
```

Connecting to the server and using the security services

The isql and bcp utilities include the following command line options to enable network-based security services on the connection:

-K *keytab_file*

-R *remote_server_principal*

-V *security_options*

-Z *security_mechanism*

Note Versions of isql and bcp for the DCE Directory Service and for DCE security services are available. They are isql_dce and bcp_dce. You must use these versions when you are using DCE.

These options are described in the following paragraphs.

-K *keytab_file* can be used only with DCE security. It specifies a DCE keytab file that contains the security key for the user logging into the server. Keytab files can be created with the DCE dcecp utility—see your DCE documentation for more information.

If the -K option is not supplied, the user of isql must be logged into DCE. If the user specifies the -U option, the name specified with -U must match the name defined for the user in DCE.

-R *remote_server_principal* specifies the principal name for the server as defined to the security mechanism. By default, a server's principal name matches the server's network name (which is specified with the -S option or the DSQUERY environment variable). The -R option must be used when the server's principal name and network name are not the same.

-V *security_options* specifies network-based user authentication. With this option, the user must log into the network's security system before running the utility. In this case, if a user specifies the -U option, the user must supply the network user name known to the security mechanism; any password supplied with the -P option is ignored.

-V can be followed by a *security_options* string of key-letter options to enable additional security services. These key letters are:

c – enable data confidentiality service

i – enable data integrity service

m – enable mutual authentication for connection establishment

o – enable data origin stamping service

r – enable data replay detection

q – enable out-of-sequence detection

-Z *security_mechanism* – specifies the name of a security mechanism to use on the connection.

Security mechanism names are defined in the *libtcl.cfg* configuration file. If no *security_mechanism* name is supplied, the default mechanism is used. For more information about security mechanism names, see the *Open Client/Server Configuration Guide* for your platform.

If you log in to the security mechanism and then log in to Adaptive Server, you do not need to specify the -U option on the utility because Adaptive Server gets the user name from the security mechanism. For example, consider the following session:

```
svrsole4% dce_login user2
Enter Password:
svrsole4% $SYBASE/bin/isql_dce -V
1> select suser_name()
2> go

------------------------------
user2
```

For this example, "user2" logs in to DCE with dce_login and then logs into Adaptive Server without specifying the -U option. The -V option without parameters implicitly specifies one security service: unified login.

For more information about Adaptive Server utilities, see the *Utility Guide*.

If you are using Client-Library to connect to Adaptive Server, you can define security properties before connecting to the server. For example, to check message sequencing, set the CS_SEC_DETECTSEQ property. For information about using security services with Client-Library, see the *Open Client Client-Library/C Reference Manual*.

Example of using security services

Assume that your login is "mary" and you want to use the DCE security mechanism with unified login (always in effect when you specify the -V option of isql_dce or bcp_dce), message confidentiality, and mutual authentication for remote procedures. You want to connect to server WOND and run remote procedures on GATEWAY with mutual authentication. Assuming that a System Security Officer has set up both WOND and GATEWAY for rpc Model B, added you as a user on both servers, and defined you as a remote, "trusted" user on GATEWAY, you can use the following process:

1 Log in to the DCE security mechanism and receive a credential:

```
dce_login mary
```

2 Log in to the Adaptive Server with isql_dce:

```
isql_dce -SWOND -Vcm
```

3 Run:

```
GATEWAY...sp_who
GATEWAY...mary_prc1
GATEWAY...mary_prc2
```

Now, all messages that Mary sends to the server and receives from the server will be encrypted (message confidentiality), and when she runs remote procedures, both the WOND and GATEWAY servers will be authenticated.

Using security mechanisms for the client

Adaptive Server, when it is booted, determines the set of security mechanisms it supports. (See "Determining security mechanisms to support" on page 513. From the list of security mechanisms that Adaptive Server supports, it must choose the one to be used for a particular client.

If the client specifies a security mechanism (for example with the -Z option of isql_dce), Adaptive Server uses that security mechanism. Otherwise, it uses the first security mechanism listed in the *libtcl.cfg* file.

Getting information about available security services

Adaptive Server enables you to:

- Determine what security mechanisms and services are supported by Adaptive Server
- Determine what security services are active for the current session
- Determine whether a particular security service is enabled for the session

Determining supported security services and mechanisms

A system table, syssecmechs, provides information about the security mechanisms and security services supported by Adaptive Server. The table, which is dynamically built when you query it, contains these columns:

- sec_mech_name is the name of the security mechanism; for example, the security mechanism might be “dce” or “NT LANMANAGER.”
- available_service is the name of a security service supported by the security mechanism; for example, the security service might be “unified login.”

Several rows may be in the table for a single security mechanism: one row for each security service supported by the mechanism.

To list all the security mechanisms and services supported by Adaptive Server, run this query:

```
select * from syssecmechs
The result might look something like:
sec_mech_name                  available_service
------------------------------ --------------------
 dce                            unifiedlogin
 dce                            mutualauth
 dce                            delegation
 dce                            integrity
 dce                            confidentiality
 dce                            detectreplay
 dce                             detectseq
```

Determining enabled security services

To determine which security services are enabled for the current session, use the function show_sec_services. For example:

```
    show_sec_services()
-----------------------------------------------------------------
          unifiedlogin mutualauth confidentiality
(1 row affected)
```

Determining whether a security service Is enabled

To determine whether a particular security service, such as "mutualauth" is enabled, use the function is_sec_service_on. The syntax is:

is_sec_service_on(*security_service_nm*)

where *security_service_nm* is a security service that is available. Use the name that is displayed when you query syssecmechs.

For example, to determine whether "mutualauth" is enabled, execute:

```
select is_sec_service_on("mutualauth")
-----------
          1

(1 row affected)
```

A result of 1 indicates the security service is enabled for the session. A result of 0 indicates the service is not in use.